JUNOS™ Internet Software Configuration Guide

Network Management

Release 5.6

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-008939-01, Revision 1

JUNOS Internet Software Configuration Guide: Network Management, Release 5.6

Writer: Carrie L. Unger, John Gilbert Chan
Editor: Stella Hackell, Ed Harper
Covers and template design: Edmonds Design

Revision History
27 December 2002—First Edition.

The information in this document is current as of the date listed in the revision history.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

Products made or sold by Juniper Networks (including the M5, M10, M20, M40, M40e, and M160 routers, T320 router, T640 routing node, and the JUNOS software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,333,650, 6,359,479, and 6,406,312.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

Table of Contents

List of Tables

List of Tables

About This Manual

This chapter provides a high-level overview of the *JUNOS Internet Software Configuration Guide: Network Management*:

Objectives

This manual provides an overview of the network management features of the JUNOS Internet software and describes how to manage your network with the JUNOS Internet software.

This manual documents the network management features in Release 5.6 of the JUNOS Internet software. To obtain the most current version of this manual and the most current version of the software release notes, refer to the product documentation page on the Juniper Networks Web site, which is located at http://www.juniper.net/.

To order printed copies of this manual or to order a documentation CD-ROM, which contains this manual, please contact your sales representative.

Audience

This manual is designed for network administrators who are configuring a Juniper Networks router. It assumes that you have a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration and management. It also assumes some knowledge of SNMP and SNMP management software.

Document Organization

- Part 1, “Network Management Introduction,” provides an overview of network management, its main components, and how you can manage your Juniper Networks routers using the JUNOS Internet software.
  - Chapter 1, “Network Management Overview,” introduces the concept of network management and its main components: fault management, configuration management, accounting management, performance management, and security management.
  - Chapter 2, “Complete Network Management Configuration Statements,” lists the complete statement hierarchies for the statements discussed in this manual. For a complete list of all configuration mode statements and commands, see the *JUNOS Internet Software Configuration Guide: Getting Started*.
- Part 2, “SNMP,” describes SNMP, how to configure the SNMP agent on the router, how to use SNMP remote operations, and lists the full definitions for Juniper Networks Enterprise MIBs and traps.
  - Chapter 3, “SNMP Overview,” provides an overview of SNMP, the protocol that allows you to manage a router running the JUNOS software, and how it is implemented in the JUNOS software.
  - Chapter 4, “Configure SNMP,” describes how to configure the SNMP agent on the router.
  - Chapter 5, “SNMP Remote Operations,” describes how to use the ping and traceroute MIBs to monitor your SNMP network remotely.
  - Chapter 6, “Juniper Networks Enterprise-Specific MIBs,” lists the full definitions of enterprise-specific MIBs supported by the JUNOS software.
  - Chapter 7, “Juniper Networks Enterprise-Specific SNMP Traps,” describes the enterprise-specific traps supported by the JUNOS software and lists their full definitions.
  - Chapter 8, “Standard SNMP Traps,” describes the standard traps supported by the JUNOS software and lists their full definitions.
  - Chapter 9, “Summary of SNMP Configuration Statements,” explains each of the SNMP configuration statements.
- Part 3, “RMON Alarms and Events” describes RMON events and alarms, how to configure them on your router, and how to use the Juniper Networks enterprise-specific RMON alarm and event MIB to monitor other MIB variables.
  - Chapter 10, “Configure RMON Alarms and Events,” provides an introduction to the RMON alarm and event features supported by the JUNOS software. It is also describes how to configure alarm and event entries and their attributes.
  - Chapter 11, “Monitor RMON Alarms and Events,” describes how to set, monitor, and poll events and alarms for any MIB object.
  - Chapter 12, “Summary of RMON Alarm and Event Configuration Statements,” explains each of the RMON alarm and event configuration statements.

- Part 4, “Interpret the Juniper Networks Enterprise-Specific MIBs,” explains how to interpret the Juniper Networks enterprise-specific MIBs.
  - Chapter 13, “Interpret the Chassis MIB,” describes each part of the Juniper Networks enterprise-specific chassis MIB.
  - Chapter 14, “Interpret the Destination Class Usage MIB,” describes each part of the Juniper Networks enterprise-specific destination class usage MIB.
  - Chapter 15, “Interpret the Ping MIB,” describes each part of the Juniper Networks enterprise-specific extensions to the ping MIB.
  - Chapter 16, “Interpret the Traceroute MIB,” describes each part of the Juniper Networks enterprise-specific extensions to the traceroute MIB.
  - Chapter 17, “Interpret the RMON Events and Alarms MIB,” describes each part of the Juniper Networks enterprise-specific extensions to the RMON alarm and event MIB.
  - Chapter 18, “Interpret the Reverse Path Forwarding MIB,” describes each part of the Juniper Networks enterprise-specific reverse path forwarding MIB.
  - Chapter 19, “Interpret the Source Class Usage MIB,” describes each part of the Juniper Networks enterprise-specific source class usage MIB.
  - Chapter 20, “Interpret the Passive Monitoring MIB,” describes each part of the Juniper Networks enterprise-specific passive monitoring MIB.
  - Chapter 21, “Interpret the SONET/SDH Interface Management MIB,” describes each part of the Juniper Networks enterprise-specific SONET/SDH MIB.
- Part 5, “Accounting Options,” describes how to configure accounting options for interfaces, firewall filters, destination classes, and the Routing Engine.
  - Chapter 22, “Accounting Options Overview,” provides background information for configuring accounting options.
  - Chapter 23, “Configure Accounting Options,” describes how to configure accounting options.
  - Chapter 24, “Summary of Accounting Options Configuration Statements,” explains each of the accounting options statements.
- Part 6, “Appendix,” includes a glossary.
  - Appendix A, “Glossary”, provides a list of terms and definitions.

This manual also contains a complete index and an index of statements and commands.

Using the Indexes

This manual contains two indexes: a complete index, which contains all index entries, and an index that contains only statements and commands.

In the complete index, bold page numbers point to pages in the statement summary chapters. The index entry for each configuration statement always contains at least two entries. The first, with a bold page number on the same line as the statement name, references the statement summary section. The second entry, "usage guidelines," references the section in a configuration guidelines chapter that describes how to use the statement.

Documentation Conventions

General Conventions

This manual uses the following text conventions:

- Statements, commands, filenames, directory names, IP addresses, and configuration hierarchy levels are shown in a sans serif font. In the following example, *stub* is a statement name and [edit protocols ospf area *area-id*] is a configuration hierarchy level:

  To configure a stub area, include the stub statement at the [edit protocols ospf area *area-id*] hierarchy level:

- In examples, text that you type literally is shown in bold. In the following example, you type the word *show*:

  [edit protocols ospf area *area-id*]
  cli# **show**
  stub <default-metric *metric*>

- Examples of command output are generally shown in a fixed-width font to preserve the column alignment. For example:

```
> show interfaces terse
Interface       Admin Link Proto Local            Remote
at-1/3/0        up    up
at-1/3/0.0      up    up   inet  1.0.0.1          --> 1.0.0.2
                           iso
fxp0            up    up
fxp0.0          up    up   inet  192.168.5.59/24
```

Conventions for Software Commands and Statements

When describing the JUNOS software, this manual uses the following type and presentation conventions:

- Statement or command names that you type literally are shown non italicized. In the following example, the statement name is *area*:

  You configure all these routers by including the following area statement at the [edit protocols ospf] hierarchy level:

- Options, which are variable terms for which you substitute appropriate values, are shown in italics. In the following example, *area-id* is an option. When you type the area statement, you substitute a value for *area-id*.

  area *area-id*;

- Optional portions of a configuration statement are enclosed in angle brackets. In the following example, the "default-metric *metric*" portion of the statement is optional:

  stub <default-metric *metric*>;

- For text strings separated by a pipe (|), you must specify either *string1* or *string2*, but you cannot specify both or neither of them. Parentheses are sometimes used to group the strings.

  string1 | *string2*
  (*string1* | *string2*)

  In the following example, you must specify either broadcast or multicast, but you cannot specify both:

  broadcast | multicast

- For some statements, you can specify a set of values. The set must be enclosed in square brackets. For example:

  community *name* members [*community-id*]

- The configuration examples in this manual are generally formatted in the way that they appear when you issue a show command. This format includes braces ({ }) and semicolons. When you type configuration statements in the CLI, you do not type the braces and semicolons. However, when you type configuration statements in an ASCII file, you must include the braces and semicolons. For example:

```
[edit]
cli# set routing-options static route default nexthop address retain
[edit]
cli# show
routing-options {
    static {
        route default {
            nexthop address;
            retain;
        }
    }
}
```

- Comments in the configuration examples are shown either preceding the lines that the comments apply to, or more often, on the same line. When comments appear on the same line, they are preceded by a pound sign (#) to indicate where the comment starts. In an actual configuration, comments can only precede a line; they cannot be on the same line as a configuration statement. For example:

```
protocols {
    mpls {
        interface (interface-name | all);    # Required to enable MPLS on the interface
    }
    rsvp {                                   # Required for dynamic MPLS only
        interface interface-name;
    }
}
```

- The general syntax descriptions provide no indication of the number of times you can specify a statement, option, or keyword. This information is provided in the text of the statement summary.

List of Technical Publications

Table 1 lists the software and hardware books for Juniper Networks routers and describes the contents of each book.

Table 1: Juniper Networks Technical Documentation

Book	Description
JUNOS Internet Software Configuration Guides	
Getting Started	Provides an overview of the JUNOS Internet software and describes how to install and upgrade the software. This manual also describes how to configure system management functions and how to configure the chassis, including user accounts, passwords, and redundancy.
Interfaces and Class of Service	Provides an overview of the interface and class-of-service functions of the JUNOS Internet software and describes how to configure the interfaces on the router.
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, and forwarding options.
Routing and Routing Protocols	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
VPNs	Provides an overview of Layer 2 and Layer 3 Virtual Private Networks (VPNs), describes how to configure VPNs, and provides configuration examples.
JUNOS Internet Software References	
Operational Mode Command Reference: Interfaces	Describes the JUNOS Internet software operational mode commands you use to monitor and troubleshoot Juniper Networks routers.
Operational Mode Command Reference: Protocols, Class of Service, Chassis, and Management	Describes the JUNOS Internet software operational mode commands you use to monitor and troubleshoot Juniper Networks routers.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.

Book	Description
JUNOScript API Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript API to monitor and configure Juniper Networks routers.
JUNOScript API Reference	Provides a reference page for each tag in the JUNOScript API.
JUNOS Internet Software Comprehensive Index	
Comprehensive Index	Provides a complete index of all JUNOS Internet software books and the *JUNOScript API Guide*.
Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routers and router components. Each router platform (M5 and M10 routers, M20 router, M40 router, M40e router, M160 router, T320 router, and T640 routing node) has its own hardware guide.
PIC Guide	Describes the router Physical Interface Cards (PICs). Each router platform has its own PIC guide.

Documentation Feedback

We are always interested in hearing from our customers. Please let us know what you like and do not like about the Juniper Networks documentation, and let us know of any suggestions you have for improving the documentation. Also, let us know if you find any mistakes in the documentation. Send your feedback to techpubs-comments@juniper.net.

How to Request Support

For technical support, contact Juniper Networks at support@juniper.net, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

Part 1

Network Management Introduction

Chapter 1

Network Management Overview

Once you have installed the router into your network, you need to manage the router within your network. Router management can be divided into five tasks:

- Fault management—Monitor the router; detect and fix faults.
- Configuration management—Configure router attributes.
- Accounting management—Collect statistics for accounting purposes.
- Performance management—Monitor and adjust router performance.
- Security management—Control router access and authenticate users.

The JUNOS Internet software network management features work in conjunction with an operations support system (OSS) to manage the router within the network. The JUNOS software can assist you in performing these management tasks, as described in Table 2.

Table 2: JUNOS Router Management Features

Task	JUNOS Software Feature
Fault management	Monitor and see faults using: ■ Operational mode commands—For more information on operational mode commands, see the *JUNOS Internet Software Operational Mode Command Reference.* ■ SNMP MIBs—For more information about SNMP MIBs, see “Juniper Networks Enterprise-Specific MIBs” on page 55. ■ Standard SNMP traps—For more information about standard SNMP traps, see “Standard SNMP Traps” on page 77. ■ Enterprise-specific SNMP traps—For more information about enterprise-specific traps, see “Juniper Networks Enterprise-Specific SNMP Traps” on page 59. ■ System log messages—For more information about how to configure system log messages, see the *JUNOS Internet Software Configuration Guide: Getting Started.* For more information about how to view system log messages, see the *JUNOS Internet Software System Log Messages Reference.*
Configuration management	■ Configure router attributes using the command-line interface (CLI) and the JUNOScript API. For more information on configuring the router using the CLI, see the *JUNOS Internet Software Configuration Guide: Getting Started*. For more information on configuring the router using the JUNOScript API, see the *JUNOScript API Reference* and the *JUNOScript API Guide.* ■ Configuration Management MIB—For more information about the Configuration Management MIB, see “Juniper Networks Enterprise-Specific MIBs” on page 55.

Task	JUNOS Software Feature
Accounting management	Perform the following accounting-related tasks: ■ Collect statistics for interfaces, firewall filters, destination classes, source classes, and the Routing Engine. For more information on collecting statistics, see “Configure Accounting Options” on page 233. ■ Use interface-specific traffic statistics and other counters, available in the standard interfaces MIB, Juniper Networks enterprise-specific extensions to the interfaces MIB, and media-specific MIBs, such as the enterprise-specific ATM MIB. ■ Use per ATM virtual circuit (VC) counters, available in the enterprise-specific ATM MIB. ■ Group source and destination prefixes into source classes and destination classes, respectively, and count packets for those classes. Collect destination class and source class usage statistics. For more information on classes, see “Juniper Networks Enterprise-Specific MIBs” on page 55,“Configure Class Usage Profiles” on page 248, the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*, and the *JUNOS Internet Software Configuration Guide: Policy Framework*. ■ Count packets as part of a firewall filter. For more information on firewall filter policies, see “Juniper Networks Enterprise-Specific MIBs” on page 55, and the *JUNOS Internet Software Configuration Guide: Policy Framework*. ■ Sample traffic, collect the samples, and send the collection to a host running the CAIDA cflowd utility. For more information on CAIDA and cflowd, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.
Performance management	Monitor performance in the following ways: ■ Use operational mode commands. For more information on monitoring performance using operational mode commands, see the *JUNOS Internet Software Operational Mode Command Reference*. ■ As part of a firewall filter. For more information on performance monitoring using firewall filters, see the *JUNOS Internet Software Configuration Guide: Policy Framework*. ■ Sample traffic, collect the samples, and send the samples to a host running the CAIDA cflowd utility. For more information on CAIDA and cflowd, see the *JUNOS Internet Software Configuration Guide: Policy Framework*. ■ Use the enterprise-specific class-of-service MIB. For more information on this MIB, see “Juniper Networks Enterprise-Specific MIBs” on page 55.
Security management	Assure security in your network in the following ways: ■ Control access to the router and authenticate users. For more information on access control and user authentication, see the *JUNOS Internet Software Configuration Guide: Getting Started*. ■ Control access to the router using SNMPv3 and SNMP over IPv6. For more information, see “Configure SNMPv3” on page 29 and “Trace SNMP Activity” on page 32.

Chapter 2

Complete Network Management Configuration Statements

This chapter shows the complete configuration statement hierarchy for the portions of the configuration discussed in this manual, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the JUNOS software, your current hierarchy level is shown in the banner on the line preceding the user@host# prompt.

For a list of the complete configuration statement hierarchy, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

This chapter is organized as follows:

[edit accounting-options] Hierarchy Level

```
[edit]
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        destination-classes {
            destination-class-name;
        }
        source-classes {
            source-class-name;
        }
    }
    file filename
        files number;
        size bytes;
        transfer-interval minutes;
    }
    filter-profile profile-name {
        counters {
            counter-name;
        }
        file filename;
        interval minutes;
        }
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
    routing-engine-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
}
```

[edit snmp] Hierarchy Level

```
[edit]
snmp {
    access {
        context context-name {
            description description;
            group group-name {
                model usm;
                read-view view-name;
                security-level (none | authentication | privacy);
                write-view view-name;
            }
        }
        group group-name {
            model usm;
            user [ user-names ];
        }
        user user-name {
            authentication-password authentication-password;
            authentication-type (none | md5 | sha};
            privacy-password privacy-password;
            privacy-type (none | des);
            clients {
                address restrict;
            }
        }
    }
    community community-name {
        authorization authorization;
        clients {
            address restrict;
        }
        view view-name;
    }
    contact contact;
    description description;
    engine-id {
        local engine-id;
    }
    interface [ interface-name ];
    location location;
    name name;
    traceoptions {
        file size size files number;
        flag flag;
    }
    rmon {
        alarm index {
            description description;
            falling-event-index index;
            falling-threshold integer;
            interval seconds;
            rising-event-index index;
            rising-threshold integer;
            sample-type (absolute-value | delta-value);
            startup-alarm (falling-alarm | rising-alarm | rising-or-falling alarm);
            variable oid-variable;
        }
```

```
        event index {
            community community-name;
            description description;
            sample-type type;
        }
    }
    traceoptions {
        file size size files number;
        flag flag;
    }
    trap-group group-name {
        categories [ categories ];
        destination-port <port-number>;
        targets {
            address;
        }
        version (all | v1 | v2);
    }
    trap-options {
        agent-address outgoing-interface;
        source-address address;
    }
    view view-name; {
        oid object-identifier (include | exclude)
    )
}
```

Part 2
SNMP

Chapter 3
SNMP Overview

Simple Network Management Protocol (SNMP) enables the monitoring of network devices from a central location. This chapter provides an overview of SNMP and describes how SNMP is implemented in the JUNOS software.

This chapter covers the following topics:

SNMP Architecture

The SNMP agent exchanges network management information with SNMP manager software running on a network management system (NMS), or host. The agent responds to requests for information and actions from the manager. The agent also controls access to the agent's management information base (MIB), the collection of objects that can be viewed or changed by the SNMP manager.

The SNMP manager collects information on network connectivity, activity, and events by polling managed devices.

Communication between the agent and the manager occurs in one of the following forms:

- Get, GetBulk, and GetNext requests—The manager requests information from the agent; the agent returns the information in a Get response message.
- Set requests—The manager changes the value of a MIB object controlled by the agent; the agent indicates status in a Set response message.
- Traps notification—The agent sends traps to notify the manager of significant events that occur on the network device.

Management Information Base

A MIB, or management information base, is a hierarchy of information used to define managed objects in a network device. The MIB structure is based on a tree structure, which defines a grouping of objects into related sets. Each object in the MIB is associated with an object identifier (OID), which names the object. The "leaf" in the tree structure is the actual managed object instance, which represents a resource, event, or activity that occurs in your network device.

MIBs are either standard or enterprise-specific. Standard MIBs are created by the IETF and documented in various RFCs. Depending on the vendor, many standard MIBs are delivered with the NMS software. You can also download the standard MIBs from the IETF Web site, http://www.ietf.org, and compile them into your NMS if necessary.

For a list of standard supported MIBS, see "SNMP Standards" on page 13.

Enterprise-specific MIBs are developed and supported by a specific equipment manufacturer. If your network contains devices that have enterprise-specific MIBs, you must obtain them from the manufacturer and compile them into your network management software.

For a list of Juniper Networks enterprise-specific supported MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 55.

SNMP Traps

A trap reports significant events occurring on a network device, most often errors or failures.

SNMP traps are defined in either standard or enterprise-specific MIBs. Standard traps are created by the IETF and documented in various RFCs. The standard traps are compiled into the network management software. You can also download the standard traps from the IETF Web site, http://www.ietf.org.

For more information on standard traps supported by the JUNOS software, see "Standard SNMP Traps" on page 77.

Enterprise-specific traps are developed and supported by a specific equipment manufacturer. If your network contains devices that have enterprise-specific traps, you must obtain them from the manufacturer and compile them into your network management software.

For more information on enterprise-specific traps supported by the JUNOS software, see "Juniper Networks Enterprise-Specific SNMP Traps" on page 59.

For information on system logging severity levels for SNMP traps, see "System Logging Severity Levels for SNMP Traps" on page 16.

SNMP Standards

The following standards documents define SNMP and the standard MIBs supported by the JUNOS software. RFCs can be found at http://www.ietf.org:

- IEEE, 802.3ad, *Aggregation of Multiple Link Segments* (only the objects dot3adAggMACAddress, dot3adAggAggregateOrIndividual, dot3adAggPortListPorts, and dot3adTablesLastChanged)
- RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*
- RFC 1157, *A Simple Network Management Protocol (SNMP)*
- RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments* (only isisSystem, isisMANAreaAddr, isisAreaAddr, isisSysProtSupp, isisSummAddr, isisCirc, isisCircLevel, isisPacketCount, isisISAdj, isisISAdjAreaAddr, isisAdjIPAddr, isisISAdjProtSupp, isisRa, and isisIPRA)
- RFC 1212, *Concise MIB Definitions*
- RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets*: MIB-II (supports MIB II and its SNMP version 2 derivatives, including statistic counters; IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable as defined in RFC 2096; SNMP management; interface management; SNMP version 1 Get and GetNext requests; version 2 GetBulk requests; and JUNOS-specific secured access lists)
- RFC 1215, *A Convention for Defining Traps for use with the SNMP* (only MIB II SNMP version 1 traps and version 2 notifications)
- RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*
- RFC 1850, *OSPF Version 2 Management Information Base* (except for the ospfOriginateNewLsas and ospfRxNewLsas objects, the Host Table, and the traps ospfOriginateLSA, ospfLsdbOverflow, and ospfLsdbApproachingOverflow)
- RFC 1901, *Introduction to Community-based SNMPv2*
- RFC 1905, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*
- RFC 1906, *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*
- RFC 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*
- RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*
- RFC 2012, *SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2*
- RFC 2013, *SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*

- RFC 2096, *IP Forwarding Table MIB*
- RFC 2115, *Management Information Base for Frame Relay DTEs Using SMIv2*
- RFC 2287, *Definitions of System-Level Managed Objects for Applications* (only sysApplInstallPkgTable, sysApplInstallElmtTable, sysApplElmtRunTable, and sysApplMapTable)
- RFC 2465, *Management Information Base for IP Version 6: Textual Conventions and General Group* (except for IPv6 interface statistics)
- RFC 2495, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types* (except for dsx1FarEndConfigTable, dsx1FarEndCurrentTable, dsx1FarEndIntervalTable, dsx1FarEndTotalTable, and dsx1FracTable)
- RFC 2496, *Definitions of Managed Objects for the DS3/E3 Interface Type* (except dsx3FarEndConfigTable, dsx3FarEndCurrentTable, dsx3FarEndIntervalTable, dsx3FarEndTotalTable, and dsx3FracTable)
- RFC 2515, *Definitions of Managed Objects for ATM Management* (except atmVpCrossConnectTable, atmVcCrossConnectTable, and aal5VccTable)
- RFC 2558, *Definitions of Managed Objects for the SONET/SDH Interface Type*
- RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework*
- RFC 2571, *An Architecture for Describing SNMP Management Frameworks* (read-only access)
- RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* (read-only access)
- RFC 2573, *SNMP Applications*
- RFC 2574, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
- RFC 2575, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* (read-only access)
- RFC 2576, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
- RFC 2578, *Structure of Management Information Version 2 (SMIv2)*
- RFC 2579, *Textual Conventions for SMIv2*
- RFC 2665, *Definitions of Managed Objects for the Ethernet-like Interface Types*
- RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*
- RFC 2790, *Host Resources MIB* (only the objects of the hrSystem and hrSWInstalled groups)

- RFC 2819, *Remote Network Monitoring Management Information Base* (the etherStatsTable for Ethernet interfaces only and the objects alarmTable, eventTable, and logTable)
- RFC 2863, *The Interfaces Group MIB*
- RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations* (only the objects pingCtlTable, pingResultsTable, pingProbeHistoryTable, pingMaxConcurrentRequests, traceRouteCtlTable, traceRouteResultsTable, traceRouteProbeHistoryTable, and traceRouteHopsTable)
- RFC 2932, *IPv4 Multicast Routing MIB*
- *IANAiftype Textual Convention MIB*, Internet Assigned Numbers Authority (referenced by RFC 2233, available at ftp://ftp.isi.edu/mib/ianaiftype.mib)
- *Internet Group Management Protocol (IGMP) MIB*, Internet draft draft-ietf-idmr-igmp-mib-13.txt
- *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2*, Internet draft draft-ietf-ppvpn-mpls-vpn-mib-04.txt (only mplsVpnScalars, mplsVpnVrfTable, mplsVpnVrfPerfTable, and mplsVpnVrfRouteTargetTable)
- *Protocol Independent Multicast (PIM) MIB*, Internet draft draft-ietf-idmr-pim-mib-09.txt

JUNOS SNMP Agent Features

The JUNOS SNMP agent software consists of an SNMP master agent that delegates all SNMP requests to subagents. Each subagent is responsible for the support of a specific set of MIBs.

The JUNOS software supports the following versions of SNMP:

- SNMPv1—The initial implementation of SNMP that defines architecture and framework for SNMP.
- SNMPv2c—The revised protocol, with improvements to performance and manager-to-manager communications. Specifically, SNMPv2c implements community strings, which act as passwords when determining who, what, and how the SNMP clients can access the data in SNMP agent. The community string is contained in SNMP Get, GetBulk, GetNext, and Set requests. The agent may require a different community string for Get, GetBulk, and GetNext requests (read-only access) than it does for Set requests (read-write access).
- SNMPv3—The most up-to-date protocol focuses on security. SNMPv3 defines a security model, User-based Security Model (USM), and an access control model, View-based Access Control Model (VACM). SNMPv3 USM provides data integrity, data origin authentication, message replay protection, and protection against disclosure of the message payload. SNMPv3 VACM provides access control to determine whether a specific type of access (read or write) to the management information is allowed.

In addition, the JUNOS SNMP agent software accepts IPv4 and IPv6 addresses for transport over IPv4 and IPv6. For IPv6, the JUNOS software supports the following IPv6 over SNMP:

- SNMP data over IPv6 networks
- IPv6-specific MIB data
- SNMP agents for IPv6

System Logging Severity Levels for SNMP Traps

For some traps, when a trap condition occurs, regardless of whether the SNMP agent sends a trap to an NMS, the trap is logged if the system logging is configured to log an event with that system logging severity level. For more information about system logging severity levels, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

For more information on system logging severity levels for standard traps, see “Standard SNMP Traps” on page 77. For more information on system logging severity levels for enterprise-specific traps, see “Juniper Networks Enterprise-Specific SNMP Traps” on page 59.

Chapter 4
Configure SNMP

To configure SNMP, include the following statements at the [edit snmp] hierarchy level of the configuration.

```
snmp {
    access {
        context context-name {
            description description;
            group group-name {
                model usm;
                read-view view-name;
                security-level (none | authentication | privacy);
                write-view view-name;
            }
        }
        group group-name {
            model usm;
            user [ user-names ];
        }
        user user-name {
            authentication-password authentication-password;
            authentication-type (none | md5 | sha};
            privacy-password privacy-password;
            privacy-type (none | des);
            clients {
                address restrict;
            }
        }
    }
    community community-name {
        authorization authorization;
        clients {
            address restrict;
        }
        view view-name;
    }
    contact contact;
    description description;
    engine-id {
        local engine-id;
    }
    interface [ interface-names ];
    location location;
    name name;
    traceoptions {
        file size size files number;
        flag flag;
    }
```

```
rmon {
    alarm index {
        description text-description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        rising-threshold integer;
        sample-type type;
        startup-alarm alarm;
        variable oid-variable;
    }
    event index {
        community community-name;
        description text-description;
        sample-type (for RMON Events) type;
    }
}
traceoptions {
    file size size files number;
    flag flag;
}
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
        address;
    }
    version (all | v1 | v2);
}
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
view view-name; {
    oid object-identifier (include | exclude)
)
}
```

For information about configuring Remote Monitoring (RMON) alarms and events, see “Configure RMON Alarms and Events” on page 115 and “Summary of RMON Alarm and Event Configuration Statements” on page 133.

By default, SNMP is disabled.

This chapter describes the minimum required configuration and discusses the following tasks for configuring SNMP:

Minimum SNMP Configuration

To configure the minimum requirements for SNMP, include statements at the [edit snmp] hierarchy level of the configuration:

```
[edit]
snmp {
    community public;
}
```

The community defined here as public grants read access to all MIB data to any client.

Configure the System Contact

You can specify an administrative contact for each system being managed by SNMP. This name is placed into the MIB II sysContact object. To configure a contact name, include the contact statement at the [edit snmp] hierarchy level:

```
[edit snmp]
contact contact;
```

If the name contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Contact

Define the system contact:

```
[edit]
snmp {
    contact "Junipero Berry, (650) 555-1234";
}
```

Configure the System Location

You can specify the location of each system being managed by SNMP. This string is placed into the MIB II sysLocation object. To configure a system location, include the location statement at the [edit snmp] hierarchy level:

```
[edit snmp]
location location;
```

If the location contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Location

Specify where the system is located:

```
[edit]
snmp {
    location "Row 11, Rack C";
}
```

Configure the System Description

You can specify a description for each system being managed by SNMP. This string is placed into the MIB II sysDescription object. To configure a description, include the description statement at the [edit snmp] hierarchy level:

```
[edit snmp]
description description;
```

If the description contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Description

Specify the system description:

```
[edit]
snmp {
    description "M40 router with 8 FPCs";
}
```

Configure the System Name

Specify the system name override:

```
[edit snmp]
name name;
```

If the name contains spaces, enclose it in quotation marks (" ").

Example: Configure the System Name

Specify the system name override:

```
[edit]
snmp {
    name "snmp 1";
}
```

Configure the SNMP Community String

The SNMP community string defines the relationship between an SNMP server system and the client systems. This string acts like a password to control the clients' access to the server. To configure a community string, include the community statement at the [edit snmp] hierarchy level:

```
[edit snmp]
community name {
    authorization authorization;
    clients {
        default restrict;
        address restrict;
    }
    view view-name;
}
```

If the community name contains spaces, enclose it in quotation marks (" ").

The default authorization level for a community is read-only. To allow Set requests within a community, you need to define that community as authorization read-write. For Set requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the view statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information on the view statement, see "view" on page 111.

The clients statement lists the IP addresses of the clients (community members) that are allowed to use this community. If no clients statement is present, all clients are allowed. For *address*, you must specify an IPv4 or IPv6 address, not a hostname. Include the default restrict option to deny access to all SNMP clients for which access is not explicitly granted. We recommend that you always include the default restrict option to limit SNMP client access to the local router.

Examples: Configure the SNMP Community String

Grant read-only access to all clients. With the following configuration, the system responds to SNMP Get, GetNext, and GetBulk requests that contain the community string public:

```
[edit]
snmp {
    community public {
        authorization read-only;
    }
}
```

Grant all clients read-write access to ping MIB and jnxPingMIB. With the following configuration, the system responds to SNMP Get, GetNext, GetBulk, and Set requests that contain the community string private and specify an OID contained in the ping MIB or jnxPingMIB hierarchy:

```
[edit]
snmp {
    view ping-mib-view {
        oid .1.3.6.1.2.1.80 include;          # pingMIB
        oid .1.3.6.1.4.1.2636.3.7 include;  # jnxPingMIB
    community private {
        authorization read-write;
        view ping-mib-view;
        }
    }
}
```

The following configuration allows read-only access to clients with IP addresses in the range 1.2.3.4/24, and denies access to systems in the range of fe80::1:2:3:4/64:

```
[edit]
snmp {
    community field-service {
        authorization read-only;
        clients {
            default restrict;                   # Restrict access to all SNMP clients not explicitly
                                                # listed on the following lines.
            1.2.3.4/24;                         # Allow access by all clients in 1.2.3.4/24; except
            fe80::1:2:3:4/64 restrict;          # fe80::1:2:3:4/64
        }
    }
}
```

Configure SNMP Trap Options and Groups

Some carriers have more than one trap receiver that forwards traps to a central NMS. This allows for more than one path for SNMP traps from a router to the central NMS through different trap receivers. A router can be configured to send the same copy of each SNMP trap to every trap receiver configured in the trap group.

The source address in the IP header of each SNMP trap packet is set to the address of the outgoing interface by default. When a trap receiver forwards the packet to the central NMS, the source address is preserved. The central NMS, looking only at the source address of each SNMP trap packet, assumes that each SNMP trap came from a different source.

In reality, the SNMP traps came from the same router, but each left the router through a different outgoing interface.

The statements discussed in the following sections are provided to allow the NMS to recognize the duplicate traps and to distinguish SNMPv1 traps based on the outgoing interface.

To configure SNMP trap options and trap groups, include the trap-options and trap-group statements at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
        address;
    }
    version (all | v1 | v2);
}
```

This section includes the following topics:

Configure SNMP Trap Options

Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router to a single address regardless of the outgoing interface. In addition, you can set the agent address of the SNMPv1 traps. For more information on the contents of SNMPv1 traps, see RFC 1157.

To configure SNMP trap options, include the trap-options statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
```

You must also configure a trap group for the trap options to take effect. For information about trap groups, see “Configure SNMP Trap Groups” on page 25.

This section contains the following topics:

Configure the Source Address for SNMP Traps

You can configure the source address of trap packets. Currently, the only value that can be specified is lo0. The value lo0 indicates the source address of the SNMP trap packets will be set to the lowest loopback address configured at the interface lo0.

To enable and configure the source address of SNMP traps, include the source-address statement at the [edit snmp trap-options] hierarchy level:

```
[edit snmp]
trap-options {
    source-address address;
}
```

To enable and configure the loopback address, include the address statement at the [edit interfaces lo0 unit 0 family inet] hierarchy level:

```
[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address ip-address;
        }
    }
}
```

Example: Configure the Loopback Address as the Source Address of Trap Packets

To configure the loopback address and source address trap option:

```
[edit snmp]
trap-options {
    source-address lo0;
}
trap-group “urgent-dispatcher” {
    version v2;
    categories link startup;
    targets {
        192.168.10.22;
        172.17.1.2;
    }
}

[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.1/32;
            address 127.0.0.1/32;
        }
    }
}
```

In this example, the IP address 10.0.0.1 is the source address of every trap sent from this router.

Configure the Agent Address for SNMP Traps

The agent address is only available in the SNMPv1 trap packets (see RFC 1157). By default, the router's default local address is used in the agent address field of the SNMPv1 trap. To configure the agent address, include the agent-address statement at the [edit snmp trap-options] hierarchy level. Currently, the agent address can only be the address of the outgoing interface:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
}
```

Example: Configure the Outgoing Interface as the Agent Address

Configure the outgoing interface as the agent address:

```
[edit snmp]
trap-options {
    agent-address outgoing-interface;
}
trap-group "urgent-dispatcher" {
    version v1;
    categories link startup;
    targets {
        192.168.10.22;
        172.17.1.2;
    }
}
```

In this example, each SNMPv1 trap packet sent has its agent address value set to the IP address of the outgoing interface.

Configure SNMP Trap Groups

You can create and name a group of one or more types of SNMP traps and then define which systems receive the group of SNMP traps. The trap group must be configured for SNMP traps to be sent. To create an SNMP trap group, include the trap-group statement at the [edit snmp] hierarchy level:

```
[edit snmp]
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
        address;
    }
    version (all | v1 | v2);
}
```

The trap group name can be any string and is embedded in the community name field of the trap. To configure your own trap group port, include the destination-port statement. The default destination port is port 162.

Each trap group you define must have a name and one or more targets, which are the systems that receive the SNMP traps. Specify the targets by IPv4 or IPv6 address, not by hostname.

Specify the types of traps the trap group can receive in the categories statement. For information about which category traps belong to, see “Standard SNMP Traps” on page 77 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 59.

A trap group can receive the following categories:

- authentication—Authentication failures
- chassis—Chassis/environment notifications
- configuration—Configuration notifications
- link—Link-related notifications (up-down transitions, DS-3 and DS-1 line status change, IPv6 interface state change, Passive Monitoring PIC overload

  

  To send Passive Monitoring PIC overload interface traps, select the link trap category.

- remote-operations—Remote operation notifications
- rmon-alarm—Alarm for RMON events
- routing—Routing protocol notifications
- sonet-alarms—SONET/SDH alarms

  

  If you omit the SONET/SDH subcategories, all SONET/SDH trap alarm types are included in trap notifications.

  If you include SONET/SDH subcategories, only those SONET/SDH trap alarm types are include in trap notifications.

  - loss-of-light—Loss of light alarm notification
  - pll-lock—PLL lock alarm notification
  - loss-of-frame—Loss of frame alarm notification
  - loss-of-signal—Loss of signal alarm notification
  - severely-errored-frame—Severely errored frame alarm notification
  - line-ais—Line AIS alarm notification
  - path-ais—Path AIS alarm notification
  - loss-of-pointer—Loss of pointer alarm notification

- ber-defect—SONET/SDH bit error rate alarm defect notification
- ber-fault—SONET/SDH error rate alarm fault notification
- line-remote-defect-indication—Line remote defect indication alarm notification
- path-remote-defect-indication—Path remote defect indication alarm notification
- remote-error-indication—Remote error indication alarm notification
- unequipped—Unequipped alarm notification
- path-mismatch—Path mismatch alarm notification
- loss-of-cell—Loss of cell delineation alarm notification
- vt-ais—VT AIS alarm notification
- vt-loss-of-pointer—VT loss of pointer alarm notification
- vt-remote-defect-indication—VT remote defect indication alarm notification
- vt-unequipped—VT Unequipped alarm notification
- vt-label-mismatch—VT label mismatch error notification
- vt-loss-of-cell—VT Loss of cell delineation notification

- startup—System warm and cold starts
- vrrp-events—VRRP events such as new-master or authentication failures

The version statement allows you to specify the SNMP version of the traps sent to targets of the trap group. If you specify v1 only, SNMPv1 traps are sent. If you specify v2 only, SNMPv2 traps are sent. If all is specified, both an SNMPv1 and an SNMPv2 trap are sent for every trap condition. For more information on the version statement, see version on page 111.

Example: Configure SNMP Trap Groups

Set up a trap notification list named urgent-dispatcher for link and startup traps. This list is used to identify the network management hosts (1.2.3.4 and fe80::1:2:3:4) to which traps generated by the local router should be sent. The name specified for a trap group is used as the SNMP community string when the agent sends traps to the listed targets.

```
[edit]
snmp {
    trap-group "urgent-dispatcher" {
        version v2;
        categories link startup;
        targets {
            1.2.3.4;
            fe80::1:2:3:4;
        }
    }
}
```

Configure the Interfaces on Which SNMP Requests Can Be Accepted

By default, all router interfaces have SNMP access privileges. To limit the access through certain interfaces only, include the interface statement at the [edit snmp] hierarchy level:

```
[edit snmp]
interface [ interface-name ];
```

Specify the names of any logical or physical interfaces that should have SNMP access privileges. Any SNMP requests entering the router from interfaces not listed are discarded.

Example: Configure Secured Access List Checking

Grant SNMP access privileges only to devices on interfaces so-0/0/0 and at-1/0/1. The following example does this by configuring a list of logical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0.0 so-0/0/0.1 at-1/0/1.0 at-1/0/1.1 ];
}
```

The following example grants the same access by configuring a list of physical interfaces:

```
[edit]
snmp {
    interface [ so-0/0/0 at-1/0/1 ];
}
```

Configure MIB Views

By default, an SNMP community grants read access and denies write access to all supported MIB objects (even communities configured as authorization read-write). To restrict or grant read or write access to a set of MIB objects, you must configure a MIB view and associate the view with a community.

To configure MIB views, include the view statement at the [edit snmp] hierarchy level:

```
[edit snmp]
view view-name {
    oid object-identifier (include | exclude) ;
}
```

The view statement defines a MIB view and identifies a group of MIB objects. Each MIB object of a view has a common OID prefix. Each object identifier represents a subtree of the MIB object hierarchy. The subtree can be represented either by a sequence of dotted integers (such as .1.3.6.1.2.1.2) or by its subtree name (such as interfaces). A configuration statement uses a view to specify a group of MIB objects on which to define access. To enable a view, you must associate the view with a community.

To remove an OID completely, use the delete view all oid *oid-number* command but omit the include parameter.

To associate MIB views with a community, include the view statement at the [edit snmp *community-name*] hierarchy level:

```
[edit snmp community community-name]
view view-name;
```

Example: Ping Proxy MIB

Restrict the ping-mib community to read and write access of the ping MIB and jnxpingMIB only. Read or write access to any other MIB using this community is not allowed.

```
[edit snmp]
view ping-mib-view {
    oid .1.3.6.1.2.1.80 include;          #pingMIB
    oid jnxPingMIB include;               #jnxPingMIB
}
community ping-mib {
    authorization read-write;
    view ping-mib-view;
}
```

For more information on the ping MIB, see RFC 2925 and “Juniper Networks Enterprise-Specific MIBs” on page 55.

Configure SNMPv3

To configure SNMPv3, you must perform the following tasks:

- Configure the Local Engine ID on page 29
- Configure SNMPv3 Access on page 30
- Configure SNMP Views on page 31

Configure the Local Engine ID

The engine ID is the administratively unique identifier for the SNMP engine. To configure the local engine ID, include the engine-id statement at the [edit snmp] hierarchy level:

```
[edit snmp]
engine-id {
    local engine-id;
}
```

The local engine ID is defined as an SNMP v3 engine's administratively unique identifier and is used for identification, not for addressing. You must configure the local engine ID explicitly. The engine ID is in text format with its fifth octet equal to 4. If the engine ID is not configured, the system default IP address of the router is used as the default engine ID. The fifth octet of the default engine ID is 1.

Configure SNMPv3 Access

SNMPv3 access sets the SNMP access levels by context, group, and user. To configure the SNMPv3 access levels, include the context, group, and user statements at the [edit snmp] hierarchy level:

```
[edit snmp]
access {
    context context-name {
        description description;
        group group-name {
            model usm;
            read-view view-name;
            security-level (none | authentication | privacy);
            write-view view-name;
        }
    }
    group group-name {
        model usm;
        user [ user-names ];
    }
    user user-name {
        authentication-password authentication-password;
        authentication-type (none | md5 | sha};
        privacy-password privacy-password;
        privacy-type (none | des);
        clients {
            address restrict;
        }
    }
}
```

The context-name statement determines what management information is accessible by an SNMP entity. An SNMP entity can have access to many access contexts and therefore requires a name to identify each context. You must also associate a context with a specific access group and configure read and write views associated with each group.

To configure access security levels, configure one of the following options for the security-level statement:

- none—No security. SNMPv3 provides no authentication and no encryption on any SNMP information.
- authentication—Provides authentication but no encryption on any SNMP information.
- privacy—Provides authentication and encryption on all SNMP information.

The group statement identifies a collection of SNMP users that share the same access policy, in which object identifiers (OIDs) are read-accessible or write-accessible. Each group is the collection of users associated with the security model. You can only specify the model usm.

The user statement identifies a person for whom management operations are performed and authorized. For each user, you can specify the authentication type, authentication password, privacy type, and privacy password. The values for authentication type are none, md5, and sha. The values for privacy type are none and des. SNMP implementations (and SNMP configuration applications) must ensure that passwords are at least 8 characters in length. All authentication and privacy passwords must be at least 8 characters.

Configure SNMP Views

SNMPv3 uses community-based views, the same as SNMPv1 and SNMPv2. For more information on how to configure community-based views, see “Configure MIB Views” on page 28.

Example: Configure SNMPv3

```
snmp {
    view all {
    oid .1.3.6.1 include;
    }
    engine-id {
        local "isp-routers-0001";
    }
    access {
        user john {
            authentication-type md5;
            authentication-password "auth-secret-password";
            privacy-type  des;
            privacy-password "priv-secret-password";
        }
    group admin {
        user john
        model usm;
        }
    context router {
        description "router context";
        group admin {
            model usm;
            security-level privacy;
            read-view all;
            write-view all;
            }
        }
    }
}
```

Trace SNMP Activity

To trace SNMP activity, include the traceoptions statement at the [edit snmp] hierarchy level:

```
[edit snmp]
traceoptions {
    file size size files number;
    flag flag;
}
```

The output of the tracing operations is placed into log files in the /var/log directory. Each of these log files is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:

- chassisd
- craftd
- ilmid
- mib2d
- rmopd
- serviced
- snmpd

You can use the file statement to control log file generation. The size statement limits the size (in kilobytes) of each log file before it is closed and compressed and a new file opened in its place. The file statement limits the total number of log files archived for each SNMP agent.

You can specify one or more of the following values for the flag option:

- all—Trace all SNMP events
- general—Trace general events
- interface-stats—Trace physical and logical interface statistics
- pdu—Trace SNMP request and response packets
- protocol-timeouts—Trace SNMP response timeouts
- routing-socket—Trace routing socket calls
- subagent—Trace subagent restarts
- timer—Trace internal timer events
- varbind-error—Trace variable binding errors

Example: Trace SNMP Activity

Trace information on SNMP packets:

```
[edit]
snmp {
    traceoptions {
        file size 10k files 5;
        flag pdu;
        flag protocol-timeouts;
        flag varbind-error;
    }
}
```

Chapter 5

SNMP Remote Operations

An *SNMP remote operation* is any process on the router that can be controlled remotely using SNMP. The JUNOS software currently provides support for two SNMP remote operations: the ping MIB and traceroute MIB, defined in RFC 2925. Using these MIBs, an SNMP client in the network management system (NMS) can:

- Start a series of operations on a router
- Receive notification when the operations are complete
- Gather the results of each operation

The JUNOS software also provides extended functionality to these MIBs in the Juniper Networks enterprise-specific extensions jnxPingMIB and jnxTraceRouteMIB. For more information about jnxPingMIB and jnxTraceRouteMIB, see “Juniper Networks Enterprise-Specific MIBs” on page 55.

This chapter covers the following topics:

SNMP Remote Operation Requirements

To use SNMP remote operations, you should be experienced with SNMP conventions. You must also configure the JUNOS software to allow the use of the remote operation MIBs.

To configure the JUNOS software for remote operations, complete the following tasks:

Set SNMP Views

All remote operation MIBs supported by the JUNOS software require that the SNMP clients have read-write privileges. The default SNMP configuration of the JUNOS software does not provide clients with a community string with such privileges.

To set read-write privileges for an SNMP community string, include the following statements at the [edit snmp] hierarchy level:

```
snmp {
    view view-name;
        oid object-identifier (include | exclude)
    }
    community community-name {
        authorization authorization;
        view view-name;
    }
}
```

Example: Set SNMP Views

To create a community named remote-community that grants SNMP clients read-write access to the ping MIB, jnxPing MIB, traceroute MIB, and jnxTraceRoute MIB, include the following statements at the [edit snmp] hierarchy level:

```
snmp {
    view remote-view {
        oid .1.3.6.1.2.1.80 include;              # pingMIB
        oid .1.3.6.1.4.1.2636.3.7 include;        # jnxPingMIB
        oid .1.3.6.1.2.1.81 include;              # traceRouteMIB
        oid .1.3.6.1.4.1.2636.3.8 include;        # jnxTraceRouteMIB
    }
    community remote-community {
        view remote-view;
        authorization read-write;
    }
}
```

For more information on the community statement, see “Configure the SNMP Community String” on page 21 and “community” on page 99.

For more information on the view statement, see “Configure MIB Views” on page 28 and “view” on page 111.

Set Trap Notification for Remote Operations

In addition to configuring the remote operations MIB for trap notification, you must also configure the JUNOS software. You must specify a target host for remote operations traps.

To configure trap notification for SNMP remote operations, include the categories and targets statements at the [edit snmp trap-group] hierarchy level:

```
snmp
   trap-group group-name {
      categories [ categories ];
      targets {
         address;
      }
   }
}
```

Example: Set Trap Notification for Remote Operations

Specify 172.17.12.213 as a target host for all remote operation traps:

```
snmp {
   trap-group remote-traps {
      categories remote-operations;
      targets {
         172.17.12.213;
      }
   }
}
```

For more information on trap groups, see “Configure SNMP Trap Groups” on page 25.

Use Variable Length String Indexes

All tabular objects in the remote operations MIBs supported by JUNOS are indexed by two variables of type SnmpAdminString. For more information on SnmpAdminString, see RFC 2571.

JUNOS does not handle SnmpAdminString any differently from the octet string variable type. However, the indexes are defined as variable length. When a variable length string is used as an index, the length of the string must be included as part of the OID.

Example: Set Variable Length String Indexes

To reference the pingCtlTargetAddress variable of a row in pingCtlTable where pingCtlOwnerIndex is bob and pingCtlTestName is test, use the following OID:

```
pingMIB.pingObjects.pingCtlTable.pingCtlEntry.pingCtlTargetAddress.”bob”.”test”
.1.3.6.1.2.1.80.1.2.1.4.3.98.111.98.4.116.101.115.116
```

For more information on the definition of the ping MIB, see RFC 2925.

Enable Logging

The SNMP error code returned in response to SNMP requests can only provide a generic description of the problem. The error descriptions logged by the remote operations daemon can often provide more detailed information on the problem and help you to solve the problem faster. This logging is not enabled by default. To enable logging, include the flag general statement at the [edit snmp traceoptions] hierarchy level:

```
snmp {
    traceoptions {
        flag general;
    }
}
```

For more information on traceoptions, see “Trace SNMP Activity” on page 32.

If the remote operations daemon receives an SNMP request that it cannot accommodate, the error is logged in the /var/log/rmopd file. To monitor this log file, issue the monitor start rmopd command in operational mode of the command-line interface.

Use the Ping MIB

A ping test is used to determine whether packets sent from the local host reach the designated host and are returned. If the designated host can be reached, the ping test provides the approximate round-trip time for the packets. Ping test results are stored in pingResultsTable and pingProbeHistoryTable.

RFC 2925 is the authoritative description of the ping MIB in detail and provides the ASN.1 MIB definition of the ping MIB. This section provides the necessary information to:

Start a Ping Test

Before you start a ping test, configure a ping MIB view. This allows SNMP Set requests on pingMIB. To start a ping test, create a row in pingCtlTable and set pingCtlAdminStatus to enabled. The minimum information that must be specified before setting pingCtlAdminStatus to enabled is:

pingCtlOwnerIndex	SnmpAdminString
pingCtlTestName	SnmpAdminString
pingCtlTargetAddress	InetAddress
pingCtlRowStatus	RowStatus

For all other values, defaults are chosen unless otherwise specified. pingCtlOwnerIndex and pingCtlTestName are used as the index, so their values are specified as part of the OID. To create a row, set pingCtlRowStatus to createAndWait or createAndGo on a row that does not already exist. A value of active for pingCtlRowStatus indicates that all necessary information has been supplied and the test can begin; pingCtlAdminStatus can be set to enabled. An SNMP Set request that sets pingCtlRowStatus to active will fail if the necessary information in the row is not specified or is inconsistent. For information about how to configure a view, see "Set SNMP Views" on page 36.

There are two ways to start a ping test:

- Use Multiple Set PDUs on page 39
- Use a Single Set PDU on page 39

Use Multiple Set PDUs

You can use multiple Set request PDUs (multiple PDUs, with one or more varbind each) and set the following variables in this order to start the test:

- pingCtlRowStatus to createAndWait
- All appropriate test variables
- pingCtlRowStatus to active

  The JUNOS software now verifies that all necessary information to run a test has been specified.
- pingCtlAdminStatus to enabled

Use a Single Set PDU

You can use a single Set request PDU (one PDU, with multiple varbinds) to set the following variables to start the test:

- pingCtlRowStatus to createAndGo
- All appropriate test variables
- pingCtlAdminStatus to enabled

Monitor a Running Ping Test

When pingCtlAdminStatus is successfully set to enabled, the following is done before the acknowledgement of the SNMP Set request is sent back to the client:

- pingResultsEntry is created if it does not already exist.
- pingResultsOperStatus transitions to enabled.

pingResultsTable

While the test is running, pingResultsEntry keeps track of the status of the test. The value of pingResultsOperStatus is enabled while the test is running and disabled when it has stopped.

The value of pingCtlAdminStatus remains enabled until you set it to disabled. Thus, to get the status of the test, you must examine pingResultsOperStatus.

The pingCtlFrequency variable can be used to schedule many tests for one pingCtlEntry. After a test ends normally (you did not stop the test) and pingCtlFrequency number of seconds has elapsed, the test is started again just as if you had set pingCtlAdminStatus to enabled. If you intervene at any time between repeated tests (you set pingCtlAdminStatus to disabled or pingCtlRowStatus to notInService), the repeat feature is disabled until another test is started and ends normally. A value of 0 for pingCtlFrequency indicates this repeat feature is not active.

pingResultsIpTgtAddr and pingResultsIpTgtAddrType are set to the value of the resolved destination address when the value of pingCtlTargetAddressType is dns. When a test starts successfully and pingResultsOperStatus transitions to enabled:

- pingResultsIpTgtAddr is set to null-string.
- pingResultsIpTgtAddrType is set to unknown.

pingResultsIpTgtAddr and pingResultsIpTgtAddrType are not set until pingCtlTargetAddress can be resolved to a numeric address.To retrieve these values, poll pingResultsIpTgtAddrType for any value other than unknown after successfully setting pingCtlAdminStatus to enabled.

At the start of a test, pingResultsSentProbes is initialized to 1 and the first probe is sent. pingResultsSentProbes increases by 1 each time a probe is sent.

As the test runs, every pingCtlTimeOut seconds, the following occurs:

If a reply for the last probe sent has not been received:

- pingProbeHistoryStatus for the corresponding pingProbeHistoryEntry in pingProbeHistoryTable is set to requestTimedOut.
- A pingProbeFailed trap is generated, if necessary.
- An attempt is made to send the next probe.

No more than one outstanding probe exists for each test.

For every probe, you can receive one of the following results:

- The target host acknowledges the probe with a response.
- The probe times out; there is no response from the target host acknowledging the probe.
- The probe could not be sent.

Each probe result is recorded in pingProbeHistoryTable. For more information on pingProbeHistoryTable, see pingProbeHistoryTable on page 42.

When a response is received from the target host acknowledging the current probe:

- pingResultsProbeResponses increases by 1.
- The following variables are updated:
  - pingResultsMinRtt—Minimum round-trip time.
  - pingResultsMaxRtt—Maximum round-trip time.
  - pingResultsAverageRtt—Average round-trip time.
  - pingResultsRttSumOfSquares—Sum of squares of round-trip times.
  - pingResultsLastGoodProbe—Timestamp of the last response.

Only probes that result in a response from the target host contribute to the calculation of the round-trip time (rtt) variables.

When a response to the last probe is received or the last probe has timed out, the test is complete.

pingProbeHistoryTable

An entry in pingProbeHistoryTable (pingProbeHistoryEntry) represents a probe result and is indexed by three variables:

- The first two variables, pingCtlOwnerIndex and pingCtlTestName, are the same ones used for pingCtlTable, which identifies the test.
- The third variable, pingProbeHistoryIndex, is a counter to uniquely identify each probe result.

The maximum number of pingProbeHistoryTable entries created for a given test is limited by pingCtlMaxRows. If pingCtlMaxRows is set to 0, no pingProbeHistoryTable entries will be created for that test.

Each time a probe result is determined, a pingProbeHistoryEntry is created and added to pingProbeHistoryTable. pingProbeHistoryIndex of the new pingProbeHistoryEntry is 1 greater than the last pingProbeHistoryEntry added to pingProbeHistoryTable for that test. pingProbeHistoryIndex is set to 1 if this is the first entry in the table. The same test can be run multiple times, so this index keeps growing.

If pingProbeHistoryIndex of the last pingProbeHistoryEntry added is 0xFFFFFFFF, the next pingProbeHistoryEntry added has pingProbeHistoryIndex set to 1.

The following is recorded for each probe result:

- pingProbeHistoryResponse—Time to live (ttl)
- pingProbeHistoryStatus—What happened and why
- pingProbeHistoryLastRC—Return code (rc) value of ICMP packet
- pingProbeHistoryTime—Timestamp when probe result was determined

When a probe cannot be sent, pingProbeHistoryResponse is set to 0. When a probe times out, pingProbeHistoryResponse is set to the difference between the time when the probe was discovered to be timed out and the time when the probe was sent.

When Traps are Generated

For any trap to be generated, the appropriate bit of pingCtlTrapGeneration must be set. You must also configure a trap group to receive remote operations. A trap is generated under the following conditions:

- A pingProbeFailed trap is generated every time pingCtlTrapProbeFailureFilter number of consecutive probes fail during the test.
- A pingTestFailed trap is generated when the test completes and at least pingCtlTrapTestFailureFilter number of probes failed.
- A pingTestCompleted trap is generated when the test completes and fewer than pingCtlTrapTestFailureFilter probes failed.

A probe is considered a failure when pingProbeHistoryStatus of the probe result is anything besides responseReceived.

For information about how to configure a trap group to receive remote operations, see "Configure SNMP Trap Groups" on page 25 and "Example: Set Trap Notification for Remote Operations" on page 37.

Gather Ping Test Results

You can either poll pingResultsOperStatus to find out when the test is complete or request to get a trap when the test is complete. For more information on pingResultsOperStatus, see pingResultsTable on page 40. For more information on ping MIB traps, see "When Traps are Generated" on page 43.

The statistics calculated and then stored in pingResultsTable include:

- pingResultsMinRtt—Minimum round-trip time.
- pingResultsMaxRtt—Maximum round-trip time.
- pingResultsAverageRtt—Average round-trip time.
- pingResultsProbeResponses—Number of responses received.
- pingResultsSentProbes—Number of attempts to send probes.
- pingResultsRttSumOfSquares—Sum of squares of round-trip times.
- pingResultsLastGoodProbe—Timestamp of the last response.

You can also consult pingProbeHistoryTable for more detailed information on each probe. The index used for pingProbeHistoryTable starts at 1, goes to 0xFFFFFFFF, and wraps to 1 again.

For example, if pingCtlProbeCount is 15 and pingCtlMaxRows is 5, then upon completion of the first run of this test, pingProbeHistoryTable will contain probes like those in Table 3.

Table 3: Results in pingProbeHistoryTable: After the First Ping Test

pingProbeHistoryIndex	Probe Result
11	Result of 11th probe from run 1
12	Result of 12th probe from run 1
13	Result of 13th probe from run 1
14	Result of 14th probe from run 1
15	Result of 15th probe from run 1

Upon completion of the first probe of the second run of this test, pingProbeHistoryTable will contain probes like those in Table 4.

Table 4: Results in pingProbeHistoryTable: After the First Probe of Second Test

pingProbeHistoryIndex	Probe Result
12	Result of 12th probe from run 1
13	Result of 13th probe from run 1
14	Result of 14th probe from run 1
15	Result of 15th probe from run 1
16	Result of the 1st probe from run 2

Upon completion of the second run of this test, pingProbeHistoryTable will contain probes like those in Table 5.

Table 5: Results in pingProbeHistoryTable: After the Second Ping Test

pingProbeHistoryIndex	Probe Result
26	Result of 11th probe from run 2
27	Result of 12th probe from run 2
28	Result of 13th probe from run 2
29	Result of 14th probe from run 2
30	Result of 15th probe from run 2

History entries can be deleted from the MIB in two ways:

- More history entries for a given test are added and the number of history entries exceeds pingCtlMaxRows. The oldest history entries are deleted to make room for the new ones.
- You delete the entire test by setting pingCtlRowStatus to destroy.

Stop a Ping Test

To stop an active test, set pingCtlAdminStatus to disabled. To stop the test and remove its pingCtlEntry, pingResultsEntry, and any pingHistoryEntry objects from the MIB, set pingCtlRowStatus to destroy.

Ping Variables

This section clarifies the ranges for the following variables that are not explicitly specified in the ping MIB:

- pingCtlDataSize—The value of this variable represents the total size of the payload (in bytes) of an outgoing probe packet. This payload includes the timestamp (8 bytes) that is used to time the probe. This is consistent with the definition of pingCtlDataSize (maximum value of 65507) and the standard ping application.

  If the value of pingCtlDataSize is between 0 and 8 inclusive, it is ignored and the payload is 8 bytes (the timestamp). The ping MIB assumes all probes are timed, so the payload must always include the timestamp.

  For example, if you wish to add an additional four bytes of payload to the packet, you must set pingCtlDataSize to 12.

- pingCtlDataFill—The first 8 bytes of the data segment of the packet is for the timestamp. After that, the pingCtlDataFill pattern is used in repetition. The default pattern (when pingCtlDataFill is not specified) is (00, 01, 02, 03 ... FF, 00, 01, 02, 03 ... FF, ...). The first iteration of the default pattern starts with 08.

- pingCtlMaxRows—The maximum value is 255.

- pingMaxConcurrentRequests—The maximum value is 50.

- pingCtlTable—The maximum number of entries allowed in this table is 100. Any attempt to create a 101st entry will result in a BAD_VALUE message for SNMPv1 and a RESOURCE_UNAVAILABLE message for SNMPv2.

- pingCtlTrapProbeFailureFilter and pingCtlTrapTestFailureFilter—A value of 0 for pingCtlTrapProbeFailureFilter or pingCtlTrapTestFailureFilter is not well defined by the ping MIB. If pingCtlTrapProbeFailureFilter is 0, pingProbeFailed traps will not be generated for the test under any circumstances.If pingCtlTrapTestFailureFilter is 0, pingTestFailed traps will not be generated for the test under any circumstances.

Use the Traceroute MIB

A traceroute test approximates the path packets take from the local host to the remote host.

RFC 2925 is the authoritative description of the traceroute MIB in detail and provides the ASN.1 MIB definition of the traceroute MIB. This section provides the necessary information to:

Start a Traceroute Test

Before you start a traceroute test, configure a traceroute MIB view. This allows SNMP Set requests on tracerouteMIB. To start a test, create a row in traceRouteCtlTable and set traceRouteCtlAdminStatus to enabled. You must specify at least the following before setting traceRouteCtlAdminStatus to enabled:

traceRouteCtlOwnerIndex	SnmpAdminString
traceRouteCtlTestName	SnmpAdminString
traceRouteCtlTargetAddress	InetAddress
traceRouteCtlRowStatus	RowStatus

For all other values, defaults are chosen unless otherwise specified. traceRouteCtlOwnerIndex and traceRouteCtlTestName are used as the index, so their values are specified as part of the OID. To create a row, set traceRouteCtlRowStatus to createAndWait or createAndGo on a row that does not already exist. A value of active for traceRouteCtlRowStatus indicates that all necessary information has been specified and the test can begin; traceRouteCtlAdminStatus can be set to enabled. An SNMP Set request that sets traceRouteCtlRowStatus to active will fail if the necessary information in the row is not specified or is inconsistent. For information about how to configure a view, see “Set SNMP Views” on page 36.

There are two ways to start a traceroute test:

Use Multiple Set PDUs

You can use multiple Set request PDUs (multiple PDUs, with one or more varbind each) and set the following variables in this order to start the test:

- traceRouteCtlRowStatus to createAndWait
- All appropriate test variables
- traceRouteCtlRowStatus to active

  The JUNOS software now verifies that all necessary information to run a test has been specified.
- traceRouteCtlAdminStatus to enabled

Use Single Set PDU

You can use a single Set request PDU (one PDU, with multiple varbinds) to set the following variables to start the test:

- traceRouteCtlRowStatus to createAndGo
- All appropriate test variables
- traceRouteCtlAdminStatus to enabled

Monitor a Running Traceroute Test

When traceRouteCtlAdminStatus is successfully set to enabled, the following is done before the acknowledgement of the SNMP Set request is sent back to the client:

- traceRouteResultsEntry is created if it does not already exist.
- traceRouteResultsOperStatus transitions to enabled.

traceRouteResultsTable

While the test is running, this traceRouteResultsTable keeps track of the status of the test. The value of traceRouteResultsOperStatus is enabled while the test is running and disabled when it has stopped.

The value of traceRouteCtlAdminStatus remains enabled until you set it to disabled. Thus, to get the status of the test, you must examine traceRouteResultsOperStatus.

The traceRouteCtlFrequency variable can be used to schedule many tests for one traceRouteCtlEntry. After a test ends normally (you did not stop the test) and traceRouteCtlFrequency number of seconds has elapsed, the test is started again just as if you had set traceRouteCtlAdminStatus to enabled. If you intervene at any time between repeated tests (you set traceRouteCtlAdminStatus to disabled or traceRouteCtlRowStatus to notInService), the repeat feature will be disabled until another test is started and ends normally. A value of 0 for traceRouteCtlFrequency indicates this repeat feature is not active.

traceRouteResultsIpTgtAddr and traceRouteResultsIpTgtAddrType are set to the value of the resolved destination address when the value of traceRouteCtlTargetAddressType is dns. When a test starts successfully and traceRouteResultsOperStatus transitions to enabled:

- traceRouteResultsIpTgtAddr is set to null-string.
- traceRouteResultsIpTgtAddrType is set to unknown.

traceRouteResultsIpTgtAddr and traceRouteResultsIpTgtAddrType are not set until traceRouteCtlTargetAddress can be resolved to a numeric address. To retrieve these values, poll traceRouteResultsIpTgtAddrType for any value other other than unknown after successfully setting traceRouteCtlAdminStatus to enabled.

At the start of a test, traceRouteResultsCurHopCount is initialized to traceRouteCtlInitialTtl, and traceRouteResultsCurProbeCount is initialized to 1. Each time a probe result is determined, traceRouteResultsCurProbeCount increases by 1. While the test is running, the value of traceRouteResultsCurProbeCount reflects the current outstanding probe for which results have not yet been determined.

traceRouteCtlProbesPerHop number of probes are sent for each ttl value. When the result of the last probe for the current hop is determined, provided that the current hop is not the destination hop, traceRouteResultsCurHopCount increases by 1, and traceRouteResultsCurProbeCount resets to 1.

At the start of a test, if this is the first time this test has been run for this traceRouteCtlEntry, traceRouteResultsTestAttempts and traceRouteResultsTestSuccesses are initialized to 0.

At the end of each test execution, traceRouteResultsOperStatus transitions to disabled, and traceRouteResultsTestAttempts increases by 1. If the test was successful in determining the full path to the target, traceRouteResultsTestSuccesses increases by 1, and traceRouteResultsLastGoodPath is set to the current time.

traceRouteProbeResultsTable

Each entry in traceRouteProbeHistoryTable is indexed by five variables:

- The first two variables, traceRouteCtlOwnerIndex and traceRouteCtlTestName, are the same ones used for traceRouteCtlTable and to identify the test.
- The third variable, traceRouteProbeHistoryIndex, is a counter, starting from 1 and wrapping at FFFFFFFF. The maximum number of entries is limited by traceRouteCtlMaxRows.
- The fourth variable, traceRouteProbeHistoryHopIndex, indicates which hop this probe is for (the actual ttl value). Thus, the first traceRouteCtlProbesPerHop number of entries created when a test starts have a value of traceRouteCtlInitialTtl for traceRouteProbeHistoryHopIndex.
- The fifth variable, traceRouteProbeHistoryProbeIndex, is the probe for the current hop. It ranges from 1 to traceRouteCtlProbesPerHop.

While a test is running, as soon as a probe result is determined, the next probe is sent. A maximum of traceRouteCtlTimeOut seconds elapses before a probe is marked with status requestTimedOut and the next probe is sent. There is never more than one outstanding probe per traceroute test. Any probe result coming back after a probe times out is ignored.

Each probe can:

- Result in a response from a host acknowledging the probe
- Time out with no response from a host acknowledging the probe
- Fail to be sent

Each probe status is recorded in traceRouteProbeHistoryTable with traceRouteProbeHistoryStatus set accordingly.

Probes that result in a response from a host record the following data:

- traceRouteProbeHistoryResponse—Round-trip time (rtt)
- traceRouteProbeHistoryHAddrType—The type of HAddr (next argument)
- traceRouteProbeHistoryHAddr—The address of the hop

All probes, regardless of whether a response for the probe is received, have the following recorded:

- traceRouteProbeHistoryStatus—What happened and why
- traceRouteProbeHistoryLastRC—Return code (rc) value of the ICMP packet
- traceRouteProbeHistoryTime—Timestamp when the probe result was determined

When a probe cannot be sent, traceRouteProbeHistoryResponse is set to 0. When a probe times out, traceRouteProbeHistoryResponse is set to the difference between the time when the probe was discovered to be timed out and the time when the probe was sent.

traceRouteHopsTable

Entries in traceRouteHopsTable are indexed by three variables:

- The first two, traceRouteCtlOwnerIndex and traceRouteCtlTestName, are the same ones used for traceRouteCtlTable and identify the test.
- The third variable, traceRouteHopsHopIndex, indicates the current hop, which starts at 1 (not traceRouteCtlInitialTtl).

When a test starts, all entries in traceRouteHopsTable with the given traceRouteCtlOwnerIndex and traceRouteCtlTestName are deleted. Entries in this table are only created if traceRouteCtlCreateHopsEntries is set to true.

A new traceRouteHopsEntry is created each time the first probe result for a given ttl is determined. The new entry is created whether or not the first probe reaches a host. The value of traceRouteHopsHopIndex is increased by 1 for this new entry.


Note

Any traceRouteHopsEntry can lack a value for traceRouteHopsIpTgtAddress if there are no responses to the probes with the given ttl.

Each time a probe reaches a host, the IP address of that host is available in the probe result. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is not set, then the value of traceRouteHopsIpTgtAddress is set to this IP address. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is the same as the IP address, then the value does not change. If the value of traceRouteHopsIpTgtAddress of the current traceRouteHopsEntry is different from this IP address, indicating a path change, a new traceRouteHopsEntry is created with:

- traceRouteHopsHopIndex variable increased by 1
- traceRouteHopsIpTgtAddress set to the IP address


Note

A new entry for a test is added to traceRouteHopsTable each time a new ttl value is used or the path changes. Thus, the number of entries for a test may exceed the number of different ttl values used.

When a probe result is determined, the value traceRouteHopsSentProbes of the current traceRouteHopsEntry increases by 1. When a probe result is determined, and the probe reaches a host:

- The value traceRouteHopsProbeResponses of the current traceRouteHopsEntry is increased by 1.
- The following variables are updated:
  - traceRouteResultsMinRtt—Minimum round-trip time.
  - traceRouteResultsMaxRtt—Maximum round-trip time.
  - traceRouteResultsAverageRtt—Average round-trip time.

- traceRouteResultsRttSumOfSquares—Sum of squares of round-trip times.
- traceRouteResultsLastGoodProbe—Timestamp of the last response.

Only probes that reach a host affect the round-trip time values.

When Traps Are Generated

For any trap to be generated, the appropriate bit of traceRouteCtlTrapGeneration must be set. You must also configure a trap group to receive remote operations. Traps are generated under the following conditions:

- traceRouteHopsIpTgtAddress of the current probe is different from the last probe with the same TTL value (traceRoutePathChange).
- A path to the target could not be determined (traceRouteTestFailed).
- A path to the target was determined (traceRouteTestCompleted).

For information about how to configure a trap group to receive remote operations, see "Configure SNMP Trap Groups" on page 25 and "Example: Set Trap Notification for Remote Operations" on page 37.

Monitor Traceroute Test Completion

When a test is complete, traceRouteResultsOperStatus transitions from enabled to disabled. This transition occurs in the following situations:

- The test ends successfully. A probe result indicates that the destination has been reached. In this case, the current hop is the last hop. The rest of the probes for this hop are sent. When the last probe result for the current hop is determined, the test ends.
- traceRouteCtlMaxTtl threshold is exceeded. The destination is never reached. The test ends after the number of probes with ttl value equal to traceRouteCtlMaxTtl have been sent.
- traceRouteCtlMaxFailures threshold is exceeded. The number of consecutive probes that end with status requestTimedOut exceeds traceRouteCtlMaxFailures.
- You end the test. You set traceRouteCtlAdminStatus to disabled or delete the row by setting traceRouteCtlRowStatus to destroy.
- You misconfigured the traceroute test. A value or variable you specified in traceRouteCtlTable is incorrect and will not allow a single probe to be sent. Because of the nature of the data, this error could not be determined until the test was started; that is, until after traceRouteResultsOperStatus transitioned to enabled. When this occurs, one entry is added to traceRouteProbeHistoryTable with traceRouteProbeHistoryStatus set to the appropriate error code.

If traceRouteCtlTrapGeneration is set properly, either the traceRouteTestFailed or traceRouteTestCompleted trap is generated.

Gather Traceroute Test Results

You can either poll traceRouteResultsOperStatus to find out when the test is complete or request to get a trap when the test is complete. For more information on traceResultsOperStatus, see traceRouteResultsTable on page 47. For more information on traceroute MIB traps, see "When Traps Are Generated" on page 51.

Statistics are calculated on a per-hop basis and then stored in traceRouteHopsTable. They include the following for each hop:

- traceRouteHopsIpTgtAddressType—Address type of host at this hop
- traceRouteHopsIpTgtAddress—Address of host at this hop
- traceRouteHopsMinRtt—Minimum round-trip time.
- traceRouteHopsMaxRtt—Maximum round-trip time.
- traceRouteHopsAverageRtt—Average round-trip time.
- traceRouteHopsRttSumOfSquares—Sum of squares of round-trip times.
- traceRouteHopsSentProbes—Number of attempts to send probes.
- traceRouteHopsProbeResponses—Number of responses received.
- traceRouteHopsLastGoodProbe—Timestamp of last response.

You can also consult traceRouteProbeHistoryTable for more detailed information on each probe. The index used for traceRouteProbeHistoryTable starts at 1, goes to 0xFFFFFFFF, and wraps to 1 again.

For example, assume the following:

- traceRouteCtlMaxRows is 10.
- traceRouteCtlProbesPerHop is 5.
- There are 8 hops to the target (the target being number 8).
- Each probe sent results in a response from a host (the number of probes sent is not limited by traceRouteCtlMaxFailures).

In this test, 40 probes are sent. At the end of the test, traceRouteProbeHistoryTable would have a history of probes like those in Table 6.

Table 6: traceRouteProbeHistoryTable

HistoryIndex	HistoryHopIndex	HistoryProbeIndex
31	7	1
32	7	2
33	7	3
34	7	4
35	7	5
36	8	1
37	8	2
38	8	3
39	8	4
40	8	5

Stop a Traceroute Test

To stop an active test, set traceRouteCtlAdminStatus to disabled. To stop a test and remove its traceRouteCtlEntry, traceRouteResultsEntry, traceRouteProbeHistoryEntry, and traceRouteProbeHistoryEntry objects from the MIB, set traceRouteCtlRowStatus to destroy.

Traceroute Variables

This section clarifies the ranges for the following variables that are not explicitly specified in the traceroute MIB:

- traceRouteCtlMaxRows—The maximum value for traceRouteCtlMaxRows is 2550. This represents the maximum ttl (255) multiplied by the maximum for traceRouteCtlProbesPerHop (10). Therefore, the traceRouteProbeHistoryTable accomodates one complete test at the maximum values for one traceRouteCtlEntry. Usually, the maximum values are not used and the traceRouteProbeHistoryTable is able to accommodate the complete history for many tests for the same traceRouteCtlEntry.

- traceRouteMaxConcurrentRequests—The maximum value is 50. If a test is running, it has one outstanding probe. traceRouteMaxConcurrentRequests represents the maximum number of traceroute tests that have traceRouteResultsOperStatus with a value of enabled. Any attempt to start a test with traceRouteMaxConcurrentRequests tests running will result in the creation of one probe with traceRouteProbeHistoryStatus set to maxConcurrentLimitReached and that test will end immediately.

- traceRouteCtlTable—The maximum number of entries allowed in this table is 100. Any attempt to create a 101st entry will result in a BAD_VALUE message for SNMPv1 and a RESOURCE_UNAVAILABLE message for SNMPv2.

Chapter 6
Juniper Networks Enterprise-Specific MIBs

The JUNOS software supports the following enterprise-specific MIBs:

- ATM MIB—Provides support for ATM interfaces and virtual connections. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-atm.txt.

- Alarm MIB—Provides support for alarms from the router. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-chassis-alarm.txt.

- Chassis MIB—Provides support for environmental monitoring (power supply state, board voltages, fans, temperatures, air flow) and inventory support for the chassis, SCB, SSB, SFM, FPCs, and PICs. For more information about the chassis MIB, see “Interpret the Chassis MIB” on page 143. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-chassis.txt.

- Class of Service MIB—Provides support for monitoring interface output queue statistics per interface and per forwarding class. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-cos.txt.

- Configuration Management MIB—Provides notification for configuration changes as SNMP traps. Each trap contains the time at which the configuration change was committed, the user who made the change, and the method by which the change was made. A history of the last 32 configuration changes is kept in jnxCmChgEventTable. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-cfgmgmt.txt.

- Destination Class Usage MIB—Provides support for monitoring packet counts based on the ingress and egress points for traffic transiting your networks. Ingress points are identified by input interface. Egress points are identified by destination prefixes grouped into one or more sets, known as destination classes. One counter is managed per interface per destination class, up to a maximum of 16 counters per interface. For information about the destination class usage MIB, see “Interpret the Destination Class Usage MIB” on page 213. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-dcu.txt.

- Firewall MIB—Provides support for monitoring firewall filter counters. Routers must have the Internet Processor II ASIC to perform firewall monitoring. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-firewall.txt.

- Interface MIB—Extends standard ifTable (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig55-net-mgmt/html/mib-jnx-if-extensions.txt.

- IPv4 MIB—Provides additional IPv4 address information, supporting the assignment of identical IPv4 addresses to separate interfaces. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ipv4.txt.

- IPv6 and ICMPv6 MIB—Provides IPv6 and ICMPv6 statistics. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ipv6.txt.

- LDP MIB—Defines LDP LSP notifications. LDP traps only support IPv4 standards. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ldp.txt.

- MPLS MIB—Provides MPLS information and defines MPLS notifications. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-mpl.txt.

- Passive Montioring MIB—Performs traffic flow monitoring and lawful interception of packets transiting between two routers. For more information about the passive monitoring MIB, see *JUNOS Internet Software Feature Guide* and “Interpret the Passive Monitoring MIB” on page 225. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-pmon.txt.

- Ping MIB— Extends the standard ping MIB control table (RFC 2925). Items in this MIB are created when entries are created in pingCtlTable of the ping MIB. Each item is indexed exactly as it is in the ping MIB. For more information about the ping MIB, see “Interpret the Ping MIB” on page 215. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ping.txt.

- RMON Events and Alarms MIB—Supports the JUNOS extensions to the standard RMON events and alarms MIB (RFC 2819). The extension augments alarmTable with additional information about each alarm. Two new traps are also defined to indicate when problems are encountered with an alarm. For more information about the RMON events and alarms MIB, see “Interpret the RMON Events and Alarms MIB” on page 219. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-rmon.txt.

- Reverse Path Forwarding MIB—Monitors statistics for traffic that is rejected because of reverse path forwarding (RPF) processing. For more information about the reverse path forwarding MIB, see “Interpret the Reverse Path Forwarding MIB” on page 221. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-rpf.txt.

- SONET/SDH Interface Management MIB—Monitors the current alarm for each SONET/SDH interface. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-sonet.txt.

- Source Class Usage MIB—Counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. For more information about the source class usage MIB, see "Interpret the Source Class Usage MIB" on page 223. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-scu.txt.

- Structure of Management Information (SMI) MIB—Explains how the Juniper Networks enterprise-specific MIBs are structured. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-smi.txt.

- Traceroute MIB—Supports the JUNOS extensions of traceroutes and remote operations. Items in this MIB are created when entries are created in the traceRouteCtlTable of the traceroute MIB. Each item is indexed exactly the same way as it is in the traceroute MIB. For more information about the traceroute MIB, see "Interpret the Traceroute MIB" on page 217. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-traceroute.txt.

Chapter 7

Juniper Networks Enterprise-Specific SNMP Traps

This chapter summarizes the enterprise-specific SNMP traps supported by the JUNOS software. For scalability reasons, the MPLS traps are generated by the ingress router only. For information on disabling the generation of MPLS traps, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

The JUNOS software supports the following enterprise-specific traps:

Juniper Networks Enterprise-Specific SNMP Version 1 Traps

The JUNOS software supports enterprise-specific version 1 traps shown in Table 7. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system log messages, see the *JUNOS Internet Software System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Table 7: Enterprise-Specific Supported SNMP Version 1 Traps

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	System Log Tag
chassis	jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2	critical	CHASSISD_SNMP_TRAP
chassis	jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2	critical	CHASSISD_SNMP_TRAP
chassis	jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5	notice	CHASSISD_SNMP_TRAP
chassis	jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3	alert	CHASSISD_SNMP_TRAP
chassis	jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1	critical	CHASSISD_SNMP_TRAP
chassis	jnxPowerSupplyOk	1.3.6.1.4.1.2636.4.2	6	1	critical	CHASSISD_SNMP_TRAP
chassis	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1	6	4	critical	CHASSISD_SNMP_TRAP
chassis	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3	alert	CHASSISD_SNMP_TRAP
rmon-alarm	jnxRmonAlarmGetFailure	1.3.6.1.4.1.2636.4.3	6	1	—	—
rmon-alarm	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3	6	2	—	—
routing	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4	6	2	—	—
routing	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4	6	1	—	—
routing	jnxMplsLspChange	1.3.6.1.4.1.2636.3.2.4	6	3	—	—
routing	jnxMplsLspDown	1.3.6.1.4.1.2636.3.2.4	6	2	—	—
routing	jnxMplsLspUp	1.3.6.1.4.1.2636.3.2.4	6	1	—	—

The sections that follow provide the Juniper Networks enterprise-specific MIBs for the SNMPv1 traps defined in Table 7, including:

- Chassis Version 1 Traps MIB on page 61
- RMON Events and Alarms Version 1 Traps MIB on page 65
- LDP Version 1 Traps MIB on page 65
- MPLS Version 1 Traps MIB on page 66

Chassis Version 1 Traps MIB

```
-- Chassis traps for chassis alarm conditions

   jnxPowerSupplyFailure                TRAP-TYPE
      ENTERPRISE jnxChassisTraps
      VARIABLES {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState
      }
      DESCRIPTION
         "A jnxPowerSupplyFailure trap signifies that
         the specified power supply in the chassis has
         been in the failure (bad DC output) condition."
   ::= 1

   jnxFanFailure                        TRAP-TYPE
      ENTERPRISE                        jnxChassisTraps
      VARIABLES {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState
      }
      DESCRIPTION
         "A jnxFanFailure trap signifies that the specified
         cooling fan or impeller in the chassis has been in
         the failure (not spinning) condition."
      ::= 2

    jnxOverTemperature                  TRAP-TYPE
      ENTERPRISE                        jnxChassisTraps
      VARIABLES {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingTemp
      }
      DESCRIPTION
         "A jnxOverTemperature trap signifies that the
         specified hardware component in the chassis has
         experienced over temperature condition."
   ::= 3
```

```
jnxRedundancySwitchover              TRAP-TYPE
   ENTERPRISE                        jnxChassisTraps
   VARIABLES {
      jnxRedundancyContentsIndex,
      jnxRedundancyL1Index,
      jnxRedundancyL2Index,
      jnxRedundancyL3Index,
      jnxRedundancyDescr,
      jnxRedundancyConfig,
      jnxRedundancyState,
      jnxRedundancySwitchoverCount,
      jnxRedundancySwitchoverTime,
      jnxRedundancySwitchoverReason
   }
   DESCRIPTION
      "A jnxRedundancySwitchover trap signifies that
      the specified hardware component in the chassis
      has experienced a redundancy switchover event
      defined as a change in state of jnxRedundancyState
      from master to backup or vice versa."
::= 4

jnxFruRemoval                        TRAP-TYPE
   ENTERPRISE                        jnxChassisTraps
   VARIABLES {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot }
   DESCRIPTION
      "A jnxFruRemoval trap signifies
      that the specified FRU (Field Replaceable Unit)
      has been removed from the chassis."
::= 5

 jnxFruInsertion                     TRAP-TYPE
   ENTERPRISE                        jnxChassisTraps
   VARIABLES {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot }
   DESCRIPTION
      "A jnxFruInsertion trap signifies that
      the specified FRU (Field Replaceable Unit) has been
      inserted into the chassis."
::= 6
```

```
jnxFruPowerOff                  TRAP-TYPE
ENTERPRISE                      jnxChassisTraps
   VARIABLES {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot,
      jnxFruOfflineReason,
      jnxFruLastPowerOff,
      jnxFruLastPowerOn }

   STATUS                       current
   DESCRIPTION
         "A jnxFruPowerOff trap signifies that the SNMPv2
         entity, acting in an agent role, has detected
         that the specified FRU (Field Replaceable Unit)
         has been powered off in the chassis."
   ::= 7

jnxFruPowerOn                   TRAP-TYPE
   ENTERPRISE                   jnxChassisTraps
   VARIABLES { jnxFruContentsIndex,
               jnxFruL1Index,
               jnxFruL2Index,
               jnxFruL3Index,
               jnxFruName,
               jnxFruType,
               jnxFruSlot,
               jnxFruOfflineReason,
               jnxFruLastPowerOff,
               jnxFruLastPowerOn }
   STATUS         current
   DESCRIPTION
         "A jnxFruPowerOn trap signifies that the SNMPv2
         entity, acting in an agent role, has detected that
         the specified FRU (Field Replaceable Unit) has been
         powered on in the chassis."
   ::= 8
```

```
-- Traps for chassis alarm cleared conditions

   jnxPowerSupplyOK                    TRAP-TYPE
      ENTERPRISE                       jnxChassisOKTraps
      VARIABLES {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState }
      DESCRIPTION
         "A jnxPowerSupplyOK trap signifies
         that the specified power supply in the
         chassis has recovered from the failure (bad DC output)
         condition."
   ::= 1

   jnxFanOK                            TRAP-TYPE
      ENTERPRISE                       jnxChassisOKTraps
      VARIABLES {
      OBJECTS        {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState }
      DESCRIPTION
         "A jnxFanOK trap signifies that
         the specified cooling fan or impeller in the chassis
         has recovered from the failure (not spinning) condition."
   ::= 2

   jnxTemperatureOK                    TRAP-TYPE
      ENTERPRISE                       jnxChassisOKTraps
      VARIABLES {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingTemp }
      DESCRIPTION
         "A jnxTemperatureOK trap signifies
         that the specified hardware component
         in the chassis has recovered from over temperature
         condition."
   ::= 3

--  ::= 4                            This OID is skipped intentionally.
```

RMON Events and Alarms Version 1 Traps MIB

```
-- define branches for jnx rmon traps
--
-- Note that we need jnxRmonTrapPrefix with the 0
-- sub-identifier to make this MIB translate to
-- an SNMPv1 format in a reversible way. For example
-- it is needed for proxies that convert SNMPv1 traps
-- to SNMPv2 notifications without MIB knowledge.
--

jnxRmonTraps                OBJECT IDENTIFIER ::= { jnxTraps 3 }
jnxRmonTrapPrefix           OBJECT IDENTIFIER ::= { jnxRmonTraps 0 }

jnxTraps                    OBJECT IDENTIFIER ::= { juniperMIB 4 }

jnxRmonTraps                OBJECT IDENTIFIER ::= { jnxTraps 3 }

jnxRmonTrapPrefix           OBJECT IDENTIFIER ::= { jnxRmonTraps 0 }

   jnxRmonAlarmGetFailure   TRAP-TYPE
      ENTERPRISE            jnxRmonTraps
      VARIABLES             { alarmIndex, alarmVariable, jnxRmonAlarmGetFailReason }
      DESCRIPTION
         "The SNMP trap that is generated when the get request for an alarm
         variable returns an error.  The specific error is identified by
         jnxRmonAlarmGetFailReason."
      ::= 1

jnxRmonGetOk                TRAP-TYPE
   ENTERPRISE               jnxRmonTraps
   VARIABLES                { alarmIndex, alarmVariable }
      DESCRIPTION
         "The SNMP trap that is generated when the get request for an alarm
         variable is successful. This is only sent after previous attempts
         were unsuccessful."
      ::= 2
```

LDP Version 1 Traps MIB

```
-LDP traps

   jnxLdpLspUp          TRAP-TYPE
      ENTERPRISE        jnxLdpTraps
      VARIABLES   {
         jnxLdpLspFec
         jnxLdpRtrid
      }
      STATUS            mandatory
      DESCRIPTION
         "The SNMP trap that is generated when an LSP comes up."
   ::= 1
```

```
jnxLdpLspDown          TRAP-TYPE
   ENTERPRISE          jnxLdpTraps
   VARIABLES   {
      jnxLdpLspFec
      jnxLdpRtrid
      jnxLdpLspDownReason
   }
   STATUS              mandatory
   DESCRIPTION
      "The SNMP trap that is generated when the LSP goes down."
   ::= 2
```

MPLS Version 1 Traps MIB

```
mplsLspUpV1                TRAP-TYPE
      ENTERPRISE           mplsTraps
      VARIABLES {
            mplsLspName,
             mplsPathName   -- LspPath
      }
   DESCRIPTION
      "An mplsLspUp trap signifies that the specified LSP is up. The current active
      path for the LSP is mplsPathName."
   ::= 1

mplsLspDownV1              TRAP-TYPE
      ENTERPRISE           mplsTraps
      VARIABLES {
          mplsLspName,
          mplsPathName     -- LspPath
      }
   DESCRIPTION
      "An mplsLspDown trap signifies that the specified LSP is down, because the current
      active path mplsPathName went down."
   ::= 2


mplsLspChangeV1            TRAP-TYPE
      ENTERPRISE           mplsTraps
      VARIABLES {
         mplsLspName,
         mplsPathName      -- toLspPath
      }
   DESCRIPTION
      "An mplsLspChange trap signifies that the the specified LSP has switched traffic to the new
      active path ‘toLspPath’. The LSP maintains up state before and after the switch over”
   ::= 3
```

Juniper Networks Enterprise-Specific SNMP Version 2 Traps

The JUNOS software supports the enterprise-specific version 2 traps shown in Table 8. The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system messages, see the *JUNOS Internet Software System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS Internet Software Configuration Guide: Getting Started.*

Table 8: Enterprise-Specific Supported SNMP Version 2 Traps

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	System Log Tag
chassis	jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	critical	CHASSISD_SNMP_TRAP
chassis	jnxFanOK	1.3.6.1.4.1.2636.4.2.2	critical	CHASSISD_SNMP_TRAP
chassis	jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	notice	CHASSISD_SNMP_TRAP
chassis	jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	notice	CHASSISD_SNMP_TRAP
chassis	jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	critical	CHASSISD_SNMP_TRAP
chassis	jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	alert	CHASSISD_SNMP_TRAP
chassis	jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	critical	CHASSISD_SNMP_TRAP
chassis	jnxRedundancySwitchOver	1.3.6.1.4.1.2636.4.1.4	critical	CHASSISD_SNMP_TRAP
chassis	jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	alert	CHASSISD_SNMP_TRAP
pmon	jnxPMonOverloadSet	1.3.6.1.4.1.2636.4.7.0.1	—	—
pmon	jnxPMonOverloadCleared	1.3.6.1.4.1.2636.4.7.0.2	—	—
rmon-alarms	jnxrisingAlarm	1.3.6.1.1.2.1.16.0.1	—	—
rmon-alarm	jnxfallingAlarm	1.3.1.2.1.16.0.2	—	—
rmon-alarm	jnxRmonAlarmGetFailure	1.3.6.1.4.1.1.2636.4.3.0.1	—	—
rmon-alarm	jnxRmonGetOk	1.3.6.1.4.1.2636.4.3.0.2	—	—
routing	jnxLdpLspDown	1.3.6.1.4.1.2636.4.4.0.2	—	—
routing	jnxLdpLspUp	1.3.6.1.4.1.2636.4.4.0.1	—	—
routing	jnxMplsLspUp	1.3.6.1.4.1.2636.3.2.4.1	—	—
routing	jnxMplsLspDown	1.3.6.1.4.1.2636.3.2.4.2	—	—
routing	jnxMplsLspChange	1.3.6.1.4.1.2636.3.2.4.3	—	—
sonet	jnxSonetAlarmSet	1.3.6.1.4.1.2636.4.6.0.1	—	—
sonet	jnxSonetAlarmCleared	1.3.6.1.4.1.2636.4.6.0.2	—	—

The sections that follow provide the Juniper Networks enterprise-specific MIBs for the SNMPv2 traps defined in Table 8, including:

Chassis Version 2 Traps MIB

```
-- Traps for chassis alarm cleared conditions

   jnxPowerSupplyOK                    NOTIFICATION-TYPE
      OBJECTS {
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState }
      STATUS                           current
      DESCRIPTION
         "A jnxPowerSupplyOK trap signifies that the
         SNMPv2 entity, acting in an agent role, has
         detected that the specified power supply in the
         chassis has recovered from the failure (bad DC output)
         condition."
      ::= { jnxChassisOKTraps 1 }

   jnxFanOK                            NOTIFICATION-TYPE
      OBJECTS{
         jnxContentsContainerIndex,
         jnxContentsL1Index,
         jnxContentsL2Index,
         jnxContentsL3Index,
         jnxContentsDescr,
         jnxOperatingState }
      STATUS                           current
      DESCRIPTION
         "A jnxFanOK trap signifies that the SNMPv2
         entity, acting in an agent role, has detected that
         the specified cooling fan or impeller in the chassis
         has recovered from the failure (not spinning) condition."
      ::= { jnxChassisOKTraps 2 }
```

```
jnxTemperatureOK                NOTIFICATION-TYPE
   OBJECTS{
      jnxContentsContainerIndex,
      jnxContentsL1Index,
      jnxContentsL2Index,
      jnxContentsL3Index,
      jnxContentsDescr,
      jnxOperatingTemp }
   STATUS                       current
   DESCRIPTION
      "A jnxTemperatureOK trap signifies that the
      SNMPv2 entity, acting in an agent role, has
      detected that the specified hardware component
      in the chassis has recovered from over temperature
      condition."
   ::= { jnxChassisOKTraps 3 }

   --  ::= { jnxChassisOKTraps 4 }        This OID is skipped intentionally.

jnxFruRemoval                   NOTIFICATION-TYPE
    OBJECTS {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot }

    STATUS                      current
    DESCRIPTION
          "A jnxFruRemoval trap signifies that the SNMPv2
          entity, acting in an agent role, has detected
          that the specified FRU (Field Replaceable Unit)
          has been removed from the chassis."
    ::= { jnxChassisTraps 5 }

jnxFruInsertion                 NOTIFICATION-TYPE
    OBJECTS  {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot }

    STATUS                       current

    DESCRIPTION
          "A jnxFruInsertion trap signifies that the SNMPv2
          entity, acting in an agent role, has detected that
          the specified FRU (Field Replaceable Unit) has been
          inserted into the chassis."
    ::= { jnxChassisTraps 6 }
```

```
jnxFruPowerOff NOTIFICATION-TYPE
    OBJECTS {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot,
      jnxFruOfflineReason,
      jnxFruLastPowerOff,
      jnxFruLastPowerOn }

    STATUS                  current
    DESCRIPTION
          "A jnxFruPowerOff trap signifies that the SNMPv2
          entity, acting in an agent role, has detected
          that the specified FRU (Field Replaceable Unit)
          has been powered off in the chassis."
    ::= { jnxChassisTraps 7 }

jnxFruPowerOn NOTIFICATION-TYPE
    OBJECTS {
      jnxFruContentsIndex,
      jnxFruL1Index,
      jnxFruL2Index,
      jnxFruL3Index,
      jnxFruName,
      jnxFruType,
      jnxFruSlot,
      jnxFruOfflineReason,
      jnxFruLastPowerOff,
      jnxFruLastPowerOn }

    STATUS                  current
    DESCRIPTION
          "A jnxFruPowerOn trap signifies that the SNMPv2
          entity, acting in an agent role, has detected that
          the specified FRU (Field Replaceable Unit) has been
          powered on in the chassis."
    ::= { jnxChassisTraps 8 }

-- Traps for chassis alarm cleared conditions

jnxPowerSupplyOK NOTIFICATION-TYPE
OBJECTS{ jnxContentsContainerIndex,
        jnxContentsL1Index,
        jnxContentsL2Index,
        jnxContentsL3Index,
        jnxContentsDescr,
        jnxOperatingState }
STATUScurrent
DESCRIPTION
   "A jnxPowerSupplyOK trap signifies that the
   SNMPv2 entity, acting in an agent role, has
   detected that the specified power supply in the
   chassis has recovered from the failure (bad DC output)
   condition."
::= { jnxChassisOKTraps 1 }
```

```
jnxFanOK NOTIFICATION-TYPE
   OBJECTS{ jnxContentsContainerIndex,
            jnxContentsL1Index,
            jnxContentsL2Index,
            jnxContentsL3Index,
            jnxContentsDescr,
            jnxOperatingState }
   STATUScurrent
   DESCRIPTION
      "A jnxFanOK trap signifies that the SNMPv2
      entity, acting in an agent role, has detected that
      the specified cooling fan or impeller in the chassis
      has recovered from the failure (not spinning) condition."
   ::= { jnxChassisOKTraps 2 }

   jnxTemperatureOK NOTIFICATION-TYPE
   OBJECTS{ jnxContentsContainerIndex,
            jnxContentsL1Index,
            jnxContentsL2Index,
            jnxContentsL3Index,
            jnxContentsDescr,
            jnxOperatingTemp }
   STATUScurrent
   DESCRIPTION
      "A jnxTemperatureOK trap signifies that the
      SNMPv2 entity, acting in an agent role, has
      detected that the specified hardware component
      in the chassis has recovered from over temperature
      condition."
   ::= { jnxChassisOKTraps 3 }
```

RMON Alarm and Event Version 2 Traps MIB

```
-- define branches for jnx rmon traps
--
-- Note that we need jnxRmonTrapPrefix with the 0
-- sub-identifier to make this MIB translate to
-- an SNMPv1 format in a reversible way. For example
-- it is needed for proxies that convert SNMPv1 traps
-- to SNMPv2 notifications without MIB knowledge.
--

jnxRmonTraps                OBJECT IDENTIFIER ::= { jnxTraps 3 }
jnxRmonTrapPrefix           OBJECT IDENTIFIER ::= { jnxRmonTraps 0 }

   jnxRmonAlarmGetFailure   NOTIFICATION-TYPE
      OBJECTS  {
         alarmIndex
         alarmVariable
         jnxRmonAlarmGetFailReason
         }
      STATUS                current
      DESCRIPTION
         "The SNMP trap that is generated when the get request for an alarm
         variable returns an error. The specific error is identified by
         jnxRmonAlarmGetFailReason."
      ::= { jnxRmonTrapPrefix 1 }
```

```
jnxRmonGetOk              NOTIFICATION-TYPE
    OBJECTS  {
        alarmIndex
        alarmVariable
        }
    STATUS                    current
    DESCRIPTION
        "The SNMP trap that is generated when the get request for an alarm
        variable is successful.  This is only sent after previous attempts
        were unsuccessful."
     ::= { jnxRmonTrapPrefix 2 }
```

LDP Version 2 Traps MIB

```
-- define branches for jnx ldp traps
--
-- Note that we need jnxLdpTrapPrefix with the 0
-- sub-identifier to make this MIB translate to
-- an SNMPv1 format in a reversible way. For example
-- it is needed for proxies that convert SNMPv1 traps
-- to SNMPv2 notifications without MIB knowledge.
--

    jnxLdpLspUp             NOTIFICATION-TYPE
        OBJECTS  {
            jnxLdpLspFec
            jnxLdpRtrid
            }
        STATUS                  current
        DESCRIPTION
            "The SNMP trap that is generated when an LSP comes up."
        ::= { jnxLdpTrapPrefix 1 }

    jnxLdpLspDown           NOTIFICATION-TYPE
        OBJECTS  {
            jnxLdpLspFec
            jnxLdpRtrid
            jnxLdpLspDownReason
            }
        STATUS                  current
        DESCRIPTION
            "The SNMP trap that is generated when the LSP goes down."
        ::= { jnxLdpTrapPrefix 2 }
```

MPLS Version 2 Traps MIB

```
-- definition of MPLS traps
--
   mplsTraps            OBJECT IDENTIFIER ::= { mpls 4 }

   mplsLspUp            NOTIFICATION-TYPE
      OBJECTS {
         mplsLspName,
         mplsPathName }   -- LspPath
      STATUS               current
      DESCRIPTION
         "An mplsLspUp trap signifies that the
         specified LSP is up. The current active
         path for the LSP is mplsPathName."
      ::= { mplsTraps 1 }

   mplsLspDown          NOTIFICATION-TYPE
      OBJECTS {
         mplsLspName,
         mplsPathName }   -- LspPath
   STATUS               current
   DESCRIPTION
      "An mplsLspDown trap signifies that the
      specified LSP is down, because the current
      active path mplsPathName went down."
      ::= { mplsTraps 2 }

   mplsLspChange        NOTIFICATION-TYPE
      OBJECTS {
         mplsLspName,
         mplsPathName }   -- toLspPath
   STATUS               current
   DESCRIPTION
      "An mplsLspChange trap signifies that the
      the specified LSP has switched traffic to
      the new active path 'toLspPath'. The LSP maintains
      up state before and after the switch over"
   ::= { mplsTraps 3 }
```

Passive Monitoring Overload Interface Version 2 Traps

To send passive monitoring overload interface traps, include the link statement at the [edit snmp trap-group categories] hierarchy level.

```
--
-- Passive Monitoring Notifications
--

jnxPMonNotificationPrefix   OBJECT IDENTIFIER ::= { jnxPMonNotifications 0 }

jnxPMonOverloadSet          NOTIFICATION-TYPE
      OBJECTS { ifDescr,
                jnxPMonLastOverload,
                jnxPMonCurrentOverload,
       jnxPMonLastOverloadDate }
      STATUS                current
      DESCRIPTION
              "Notification of a new overload condition on a Passive
               Monitoring interface."
      ::= { jnxPMonNotificationPrefix 1 }

jnxPMonOverloadCleared      NOTIFICATION-TYPE
      OBJECTS { ifDescr,
                jnxPMonLastOverload,
                jnxPMonCurrentOverload,
               jnxPMonLastOverloadDate }
      STATUS                current
      DESCRIPTION
              "Notification of a cleared overload condition on a Passive
               Monitoring interface."
      ::= { jnxPMonNotificationPrefix 2 }
```

SONET/SDH Interface Version 2 Traps

```
--
-- SONET/SDH Interface Notifications
--

jnxSonetNotificationPrefix          OBJECT IDENTIFIER ::= { jnxSonetNotifications 0 }

jnxSonetAlarmSet                    NOTIFICATION-TYPE
   OBJECTS { ifDescr,
             jnxSonetLastAlarmId,
             jnxSonetCurrentAlarms,
             jnxSonetLastAlarmDate }
   STATUS                           current
   DESCRIPTION
      "Notification of a recently set sonet alarm."
   ::= { jnxSonetNotificationPrefix 1 }

jnxSonetAlarmCleared                NOTIFICATION-TYPE
   OBJECTS { ifDescr,
             jnxSonetLastAlarmId,
             jnxSonetCurrentAlarms,
             jnxSonetLastAlarmDate }
   STATUS                           current
   DESCRIPTION
      "Notification of a recently cleared sonet alarm."
   ::= { jnxSonetNotificationPrefix 2 }
```

Chapter 8
Standard SNMP Traps

This chapter summarizes the standard SNMP traps supported by the JUNOS software. For scalability reasons, the MPLS traps are generated by the ingress router only. For information on disabling the generation of MPLS traps, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

The JUNOS software supports the following standard SNMP traps:

Standard SNMP Version 1 Traps

Table 9 provides an overview of the standard traps for SNMPv1. The traps are organized first by trap category and then by trap name and include their enterprise ID, generic trap number, and specific trap number. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information on system log messages, see the *JUNOS Internet Software System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Table 9: Standard Supported SNMP Version 1 Traps

Trap Category	Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number	System Logging Severity Level	Syslog Tag
authentication	authenticationFailure	1.3.6.1.4.1.2636	4	0	notice	SNMPD_TRAP_GEN_FAILURE
link	linkDown	1.3.6.1.4.1.2636	2	0	info	SNMP_TRAP_LINK_DOWN
link	linkUp	1.3.6.1.4.1.2636	3	0	warning	SNMP_TRAP_LINK_UP
remote-operations	pingProbeFailed	1.3.6.1.2.1.80.0	6	1	info	SNMP_TRAP_PING_PROBE_FAILED
remote-operations	pingTestFailed	1.3.6.1.2.1.80.0	6	2	info	SNMP_TRAP_PING_TEST_FAILED
remote-operations	pingTestCompleted	1.3.6.1.2.1.80.0	6	3	info	SNMP_TRAP_PING_TEST_COMPLETED
remote-operations	traceRoutePathChange	1.3.6.1.2.1.81.0	6	1	info	SNMP_TRAP_TRACE_ROUTE_PATH_ CHANGE
remote-operations	traceRouteTestFailed	1.3.6.1.2.1.81.0	6	2	info	SNMP_TRAP_TRACE_ROUTE_TEST_ FAILED
remote-operations	traceRouteTestCompleted	1.3.6.1.2.1.81.0	6	3	info	SNMP_TRAP_TRACE_ROUTE_TEST_ COMPLETED
rmon-alarm	fallingAlarm	1.3.6.1.2.1.16	6	2	—	—
rmon-alarm	risingAlarm	1.3.6.1.2.1.16	6	1	—	—
routing	bgpEstablished	1.3.6.1.2.1.15.7	6	1	—	—
routing	bgpBackwardTransition	1.3.6.1.2.1.15.7	6	2	—	—
routing	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2	6	1	—	—
routing	ospfNbrStateChange	1.3.6.1.2.1.14.16.2	6	2	—	—
routing	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2	6	3	—	—
routing	ospfIfConfigError	1.3.6.1.2.1.14.16.2	6	4	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2	6	5	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2	6	6	—	—
routing	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2	6	7	—	—
routing	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	8	—	—
routing	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2	6	9	—	—
routing	ospfTxRetransmit	1.3.6.1.2.1.14.16.2	6	10	—	—
routing	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2	6	11	—	—
routing	ospfOriginateLsa	1.3.6.1.2.1.14.16.2	6	12	—	—
routing	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2	6	13	—	—
routing	ospfLsdbOverflow	1.3.6.1.2.1.14.16.2	6	14	—	—
routing	ospfLsdbApproachingOver flow	1.3.6.1.2.1.14.16.2	6	15	—	—
routing	ospfIfStateChange	1.3.6.1.2.1.14.16.2	6	16	—	—
startup	coldStart	1.3.6.1.4.1.2636	0	0	critical	SNMPD_TRAP_COLD_START
startup	warmStart	1.3.6.1.4.1.2636	1	0	error	SNMPD_TRAP_WARM_START
vrrp	vrrpTrapNewMaster	1.3.6.1.2.1.68	6	1	warning	VRRPD_NEWMASTER_TRAP
vrrp	vrrpTrapAuthFailure	1.3.6.1.2.1.68	6	2	warning	VRRPD_AUTH_FAILURE_TRAP

SNMPv1 also supports the following standard traps:

SNMP Version 1 Standard Traps

The JUNOS software supports the standard SNMP version 1 traps, which are taken from RFC 1215, *Convention for defining traps for use with the SNMP.*

```
coldStart               TRAP-TYPE
   ENTERPRISE       snmp
   DESCRIPTION
      “A coldStart trap signifies that the sending protocol entity is reinitializing itself such
      that the agent's configuration or the protocol entity implementation may be altered.”
::= 0

warmStart               TRAP-TYPE
   ENTERPRISE       snmp
   DESCRIPTION
      “A warmStart trap signifies that the sending protocol entity is reinitializing itself such
      that neither the agent configuration nor the protocol entity implementation is altered.”
::= 1


linkDown                TRAP-TYPE
   ENTERPRISE  snmp
   VARIABLES   { ifIndex }
   DESCRIPTION
      "A linkDown trap signifies that the sending protocol entity recognizes a failure in one of
      the communication links represented in the agent's configuration."
   ::= 2

linkUp                  TRAP-TYPE
   ENTERPRISE       snmp
   VARIABLES        { ifIndex }
   DESCRIPTION
      "A linkUp trap signifies that the sending protocol entity recognizes that one of the
      communication links represented in the agent's configuration has come up."
   ::= 3

authenticationFailure TRAP-TYPE
   ENTERPRISE       snmp
   DESCRIPTION
      "An authenticationFailure trap signifies that the sending protocol entity is the addressee
      of a protocol message that is not properly authenticated.  While implementations of the
      SNMP must be capable of generating this trap, they must also be capable of suppressing the
      emission of such traps via an implementation- specific mechanism."
   ::= 4
```

```
egpNeighborLoss TRAP-TYPE
   ENTERPRISE  snmp
   VARIABLES   { egpNeighAddr }
   DESCRIPTION
      "An egpNeighborLoss trap signifies that an EGP neighbor for whom the sending protocol entity
      was an EGP peer has been marked down and the peer relationship no longer obtains."
   ::= 5
```

SNMP Version 1 Ping Traps MIB

The JUNOS software supports the SNMP traps from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*, converted to SNMPv1 format:

```
-definition of ping MIB traps

   SNMP Version 1 Traceroute Traps MIB
      pingProbeFailed TRAP-TYPE
      ENTERPRISE  pingMIB
      VARIABLES   {
         pingCtlTargetAddressType, pingCtlTargetAddress,
         pingResultsOperStatus, pingResultsIpTargetAddressType,
         pingResultsIpTargetAddress, pingResultsMinRtt,
         pingResultsMaxRtt, pingResultsAverageRtt,
         pingResultsProbeResponses, pingResultsSentProbes,
         pingResultsRttSumOfSquares, pingResultsLastGoodProbe
         }
      STATUS                          mandatory
      DESCRIPTION
         "Generated when a probe failure is detected when the
         corresponding pingCtlTrapGeneration object is set to
         probeFailure(0) subject to the value of
         pingCtlTrapProbeFailureFilter.  The object
         pingCtlTrapProbeFailureFilter can be used to specify the
         number of successive probe failures that are required
         before this notification can be generated."
   ::= 1

   pingTestFailed                  TRAP-TYPE
      ENTERPRISE                   pingMIB
      VARIABLES   {
         pingCtlTargetAddressType, pingCtlTargetAddress,
         pingResultsOperStatus, pingResultsIpTargetAddressType,
         pingResultsIpTargetAddress, pingResultsMinRtt,
         pingResultsMaxRtt, pingResultsAverageRtt,
         pingResultsProbeResponses, pingResultsSentProbes,
         pingResultsRttSumOfSquares, pingResultsLastGoodProbe
         }
      STATUS                          mandatory
      DESCRIPTION
         "Generated when a ping test is determined to have failed
         when the corresponding pingCtlTrapGeneration object is
         set to testFailure(1).  In this instance
         pingCtlTrapTestFailureFilter should specify the number of
         probes in a test required to have failed in order to
         consider the test as failed."
   ::= 2
```

```
pingTestCompleted                    TRAP-TYPE
   ENTERPRISE  pingMIB
   VARIABLES   {
      pingCtlTargetAddressType, pingCtlTargetAddress,
      pingResultsOperStatus, pingResultsIpTargetAddressType,
      pingResultsIpTargetAddress, pingResultsMinRtt,
      pingResultsMaxRtt, pingResultsAverageRtt,
      pingResultsProbeResponses, pingResultsSentProbes,
      pingResultsRttSumOfSquares, pingResultsLastGoodProbe
      }
   STATUS                            mandatory
   DESCRIPTION
      "Generated at the completion of a ping test when the
      corresponding pingCtlTrapGeneration object is set to
      testCompletion(4)."
::= 3
```

SNMP Version 1 Traceroute Traps MIB

The JUNOS software supports the SNMP traps from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*, converted to SNMPv1 format:

```
-definition of traceroute traps

   traceRoutePathChange       TRAP-TYPE
      ENTERPRISE              traceRouteMIB
      VARIABLES   {
         traceRouteCtlTargetAddressType,
         traceRouteCtlTargetAddress,
         traceRouteResultsIpTgtAddrType,
         traceRouteResultsIpTgtAddr
         }
      STATUS                  mandatory
      DESCRIPTION
         "The path to a target has changed."
   ::= 1

   traceRouteTestFailed       TRAP-TYPE
      ENTERPRISE              traceRouteMIB
      VARIABLES   {
         traceRouteCtlTargetAddressType,
         traceRouteCtlTargetAddress,
         traceRouteResultsIpTgtAddrType,
         traceRouteResultsIpTgtAddr
         }
      STATUS                  mandatory
      DESCRIPTION
      "Could not determine the path to a target."
      ::= 2
```

```
traceRouteTestCompleted   TRAP-TYPE
   ENTERPRISE             traceRouteMIB
   VARIABLES   {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
      }
   STATUS                 mandatory
   DESCRIPTION
      "The path to a target has just been determined."
::= 3
```

SNMP Version 1 VRRP Traps MIB

The JUNOS software supports the SNMP traps from RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*, converted to SNMPv1 format:

```
-definition of vrrp traps

   vrrpTrapNewMaster       TRAP-TYPE
      ENTERPRISE           vrrpMIB
      VARIABLES   {
         vrrpOperMasterIpAddr
         }
-     STATUS                mandatory
      DESCRIPTION
         "The newMaster trap indicates that the sending agent
         has transitioned to 'Master' state."
   ::= 1

   vrrpTrapAuthFailure     TRAP-TYPE
      ENTERPRISE           vrrpMIB
      VARIABLES   {
         vrrpTrapPacketSrc
         vrrpTrapAuthErrorType
         }
-     STATUS                mandatory
      DESCRIPTION
         "A vrrpAuthFailure trap signifies that a packet has
         been received from a router whose authentication key
         or authentication type conflicts with this router's
         authentication key or authentication type. Implementation
         of this trap is optional."
   ::= 2
```

Standard SNMP Version 2 Traps

Table 10 provides an overview of the standard SNMPv2 traps supported by the JUNOS software. The traps are organized first by trap category and then by trap name and include their snmpTrapOID. The system logging severity levels are listed for those traps that have them with their corresponding system log tag. For traps that do not have corresponding system logging severity levels, the cell in the table is marked with an em-dash (—).

For more information about system log messages, see the *JUNOS Internet Software System Log Messages Reference*. For more information about configuring system logging, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Table 10: Standard Supported SNMP Version 2 Traps

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag
authentication	authenticationFailure	1.3.6.1.6.3.1.1.5.5	notice	SNMPD_TRAP_GEN_FAILURE
link	linkDown	1.3.6.1.6.3.1s.1.5.4	info	SNMP_TRAP_LINK_DOWN
link	linkUp	1.3.6.1.6.3.1.1.5.4	warning	SNMP_TRAP_LINK_UP
remote-operations	pingProbeFailed	1.3.6.1.2.1.80.0.1	info	SNMP_TRAP_PING_PROBE_FAILED
remote-operations	pingTestFailed	1.3.6.1.2.1.80.0.2	info	SNMP_TRAP_PING_TEST_FAILED
remote-operations	pingTestCompleted	1.3.6.1.2.1.80.0.3	info	SNMP_TRAP_PING_TEST_COMPLETED
remote-operations	traceRoutePathChange	1.3.6.1.2.1.81.0.1	info	SNMP_TRAP_TRACE_ROUTE_PATH_ CHANGE
remote-operations	traceRouteTestFailed	1.3.6.1.2.1.81.0.2	info	SNMP_TRAP_TRACE_ROUTE_TEST_ FAILED
remote-operations	traceRouteTestCompleted	1.3.6.1.2.1.81.0.3	info	SNMP_TRAP_TRACE_ROUTE_TEST_ COMPLETED
rmon-alarm	fallingAlarm	1.3.6.1.2.1.15.7.1	—	—
rmon-alarm	risingAlarm	1.3.6.1.2.1.15.7.2	—	—
routing	bgpEstablished	.1.3.6.1.2.1.15.7.1	—	—
routing	bgpBackwardTransition	.1.3.6.1.2.1.15.7.2	—	—
routing	ospfVirtIfStateChange	1.3.6.1.2.1.14.16.2.1	—	—
routing	ospfNbrStateChange	1.3.6.1.2.1.14.16.2.2	—	—
routing	ospfVirtNbrStateChange	1.3.6.1.2.1.14.16.2.3	—	—
routing	ospfIfConfigError	1.3.6.1.2.1.14.16.2.4	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.5	—	—
routing	ospfVirtIfConfigError	1.3.6.1.2.1.14.16.2.6	—	—
routing	ospfIfAuthFailure	1.3.6.1.2.1.14.16.2.7	—	—
routing	ospfIfRxBadPacket	1.3.6.1.2.1.14.16.2.8	—	—
routing	ospfVirtIfRxBadPacket	1.3.6.1.2.1.14.16.2.9	—	—
routing	ospfTxRetransmit	1.3.6.1.2.1.14.16.2.10	—	—
routing	ospfVirtIfTxRetransmit	1.3.6.1.2.1.14.16.2.11	—	—
routing	ospfOriginateLsa	1.3.6.1.2.1.14.16.2.12	—	—
routing	ospfMaxAgeLsa	1.3.6.1.2.1.14.16.2.13	—	—
routing	ospfLsdbOverflow	1.3.6.1.2.1.14.16.2.14	—	—

Trap Category	Trap Name	snmpTrapOID	System Logging Severity Level	Syslog Tag
routing	ospfLsdbApproachingOverflow	1.3.6.1.2.1.14.16.2.15	—	—
routing	ospfIfStateChange	1.3.6.1.2.1.14.16.2.16	—	—
startup	coldStart	1.3.6.1.6.3.1.1.5.1	critical	SNMPD_TRAP_COLD_START
startup	warmStart	1.3.6.1.6.3.1.1.5.2	error	SNMPD_TRAP_WARM_START
vrrp	vrrpTrapNewMaster	1.3.6.1.2.1.68.0.1	warning	VRRPD_NEWMASTER_TRAP
vrrp	vrrpTrapAuthFailure	1.3.6.1.2.1.68.0.2	warning	VRRPD_AUTH_FAILURE_TRAP

The JUNOS software supports the following standard SNMP version 2 traps:

SNMP Version 2 Standard Traps

The JUNOS software supports the standard SNMP version traps, which are taken from RFCs 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*, and RFC 2863, *The Interfaces Group MIB*:

```
coldStart           NOTIFICATION-TYPE
   STATUS           current
   DESCRIPTION
      "A coldStart trap signifies that the SNMPv2 entity, acting
      in an agent role, is reinitializing itself and that its
      configuration may have been altered."
::= { snmpTraps 1 }

warmStart           NOTIFICATION-TYPE
   STATUS           current
   DESCRIPTION
      "A warmStart trap signifies that the SNMPv2 entity, acting
      in an agent role, is reinitializing itself such that its
      configuration is unaltered."
::= { snmpTraps 2 }

linkDown            NOTIFICATION-TYPE
   OBJECTS {
```

```
        ifIndex
        ifAdminStatus
        ifOperStatus
        }
    STATUS              current
    DESCRIPTION
        "A linkDown trap signifies that the SNMP entity, acting in
         an agent role, has detected that the ifOperStatus object for
        one of its communication links is about to enter the down
        state from some other state (but not from the notPresent
        state).  This other state is indicated by the included value
        of ifOperStatus."
::= { snmpTraps 3 }

linkUp                  NOTIFICATION-TYPE
    OBJECTS {
        ifIndex
        ifAdminStatus
        ifOperStatus
        }
    STATUS              current
    DESCRIPTION
        "A linkUp trap signifies that the SNMP entity, acting in an
        agent role, has detected that the ifOperStatus object for
        one of its communication links left the down state and
        transitioned into some other state (but not into the
        notPresent state).  This other state is indicated by the
        included value of ifOperStatus."
::= { snmpTraps 4 }

authenticationFailureNOTIFICATION-TYPE
    STATUS              current
    DESCRIPTION
        "An authenticationFailure trap signifies that the SNMPv2
        entity, acting in an agent role, has received a protocol
        message that is not properly authenticated.  While all
        implementations of the SNMPv2 must be capable of generating
        this trap, the snmpEnableAuthenTraps object indicates
        whether this trap will be generated."
::= { snmpTraps 5 }
```

SNMP Version 2 BGP Traps MIB

The JUNOS software supports the BGP standard SNMP version 2 traps. The following descriptions are taken from RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*:

```
bgpEstablished          NOTIFICATION-TYPE
   OBJECTS     {
      bgpPeerLastError
      bgpPeerState
      }
   STATUS               current
   DESCRIPTION
      "The BGP Established event is generated when the BGP FSM enters the ESTABLISHED state."
::= { bgpTraps 1 }
bgpBackwardTransitionNOTIFICATION-TYPE
   OBJECTS     {
      bgpPeerLastError
      bgpPeerState
      }
   STATUS               current
   DESCRIPTION
      "The BGPBackwardTransition Event is generated when the BGP FSM moves from a
      higher numbered state to a lower numbered state."
::= { bgpTraps 2 }
```

SNMP Version 2 OSPF Traps MIB

The JUNOS software supports the OSPF SNMP version 2 traps. The following descriptions are taken from RFC 1850, *OSPF Version 2 Management Information Base*:

```
ospfIfStateChange       NOTIFICATION-TYPE
   OBJECTS  {
      ospfRouterId, -- The originator of the trap
      ospfIfIpAddress,
                  ospfAddressLessIf,
                  ospfIfState } -- The new state
   STATUS               current
DESCRIPTION
      "An ospfIfStateChange trap signifies that there has been a change in the state of a non-virtual
      OSPF interface. This trap should be generated when the interface state regresses (e.g., goes
      from Dr to Down) or progresses to a terminal state (i.e., Point-to-Point, DR Other, Dr, or
      Backup)."
::= { ospfTraps 16 }

ospfVirtIfStateChange NOTIFICATION-TYPE
OBJECTS           {ospfRouterId, -- The originator of the trap
                  ospfVirtIfAreaId,
                  ospfVirtIfNeighbor,
                  ospfVirtIfState } -- The new state
STATUS               current
DESCRIPTION
      "An ospfIfStateChange trap signifies that there has been a change in the state of an OSPF vir-
      tual interface. This trap should be generated when the interface state regresses (e.g., goes
      from Point-to-Point to Down) or progresses to a terminal state (i.e., Point-to-Point)."
::= { ospfTraps 1 }
```

```
ospfNbrStateChange NOTIFICATION-TYPE
OBJECTS          {ospfRouterId, -- The originator of the trap
                 ospfNbrIpAddr,
                 ospfNbrAddressLessIndex,
                 ospfNbrRtrId,
                 ospfNbrState } -- The new state
   STATUS            current
   DESCRIPTION
      "An ospfNbrStateChange trap signifies that there has been a change in the state of a non-
      virtual OSPF neighbor. This trap should be generated when the neighbor state regresses
      (e.g., goes from Attempt or Full to 1-Way or Down) or progresses to a terminal state (e.g.,
       2-Way or Full). When a neighbor transitions from or to Full on non-broadcast multi-access
      and broadcast networks, the trap should be generated by the designated router. A designated
      router transitioning to Down will be noted by ospfIfStateChange."
::= { ospfTraps 2 }

ospfVirtNbrStateChange NOTIFICATION-TYPE
   OBJECTS           { ospfRouterId, -- The originator of the trap
                   ospfVirtNbrArea,
                   ospfVirtNbrRtrId,
                   ospfVirtNbrState } -- The new state
   STATUS            current
DESCRIPTION
       "An ospfIfStateChange trap signifies that there has been a change in the state of an OSPF vir-
       tual neighbor. This trap should be generated when the neighbor state regresses (e.g., goes
       from Attempt or Full to 1-Way or Down) or progresses to a terminal state (e.g., Full)."
::= { ospfTraps 3 }

ospfIfConfigError NOTIFICATION-TYPE
OBJECTS          {ospfRouterId, -- The originator of the trap
                 ospfIfIpAddress,
                 ospfAddressLessIf,
                 ospfPacketSrc, -- The source IP address
                 ospfConfigErrorType, -- Type of error
                 ospfPacketType }
STATUS             current
DESCRIPTION
   "An ospfIfConfigError trap signifies that a packet has been received on a non-virtual in-
   terface from a router whose configuration parameters conflict with this router's confi-
   guration parameters. Note that the event optionMismatch should cause a trap only if it
   prevents an adjacency from forming."
::= { ospfTraps 4 }

ospfVirtIfConfigError NOTIFICATION-TYPE
   OBJECTS           {ospfRouterId, -- The originator of the trap
                   ospfVirtIfAreaId,
                   ospfVirtIfNeighbor,
                   ospfConfigErrorType, -- Type of error
                   ospfPacketType }
   STATUS            current
   DESCRIPTION
      "An ospfConfigError trap signifies that a packet has been received on a virtual interface
      from a router whose configuration parameters conflict with this router's configuration
      parameters. Note that the event optionMismatch should cause a trap only if it prevents an ad-
      jacency from forming."
::= { ospfTraps 5 }
```

```
    ospfIfAuthFailure NOTIFICATION-TYPE
        OBJECTS         {ospfRouterId, -- The originator of the trap
                         ospfIfIpAddress,
                         ospfAddressLessIf,
                         ospfPacketSrc, -- The source IP address
                         ospfConfigErrorType, -- authTypeMismatch or
                                              -- authFailure
                         ospfPacketType }
        STATUS          current
        DESCRIPTION
            "An ospfIfAuthFailure trap signifies that a packet has been received on a non-virtual in-
            terface from a router whose authentication key or authentication type conflicts with this
            router's authentication key or authentication type."
        ::= { ospfTraps 6 }

        ospfVirtIfAuthFailure NOTIFICATION-TYPE
        OBJECTS         {ospfRouterId, -- The originator of the trap
                         ospfVirtIfAreaId,
                         ospfVirtIfNeighbor,
                         ospfConfigErrorType, -- authTypeMismatch or
                                              -- authFailure
                         ospfPacketType }
        STATUS          current
        DESCRIPTION
            "An ospfVirtIfAuthFailure trap signifies that a packet has been received on a virtual interface
            from a router whose authentication key or authentication type conflicts with this router's
            authentication key or authentication type."
        ::= { ospfTraps 7 }

        ospfIfRxBadPacket NOTIFICATION-TYPE
        OBJECTS         {ospfRouterId, -- The originator of the trap
                         ospfIfIpAddress,
                         ospfAddressLessIf,
                         ospfPacketSrc, -- The source IP address
                         ospfPacketType }
        STATUS          current
        DESCRIPTION
            "An ospfIfRxBadPacket trap signifies that an OSPF packet has been received on a non-virtual
            interface that cannot be parsed."
        ::= { ospfTraps 8 }

        ospfVirtIfRxBadPacket NOTIFICATION-TYPE
        OBJECTS         {ospfRouterId, -- The originator of the trap
                         ospfVirtIfAreaId,
                         ospfVirtIfNeighbor,
                         ospfPacketType }
    STATUS              current
    DESCRIPTION
        "An ospfRxBadPacket trap signifies that an OSPF packet has been received on a virtual interface
        that cannot be parsed."
        ::= { ospfTraps 9 }
```

```
ospfTxRetransmit NOTIFICATION-TYPE
   OBJECTS          {ospfRouterId, -- The originator of the trap
                    ospfIfIpAddress,
                    ospfAddressLessIf,
                    ospfNbrRtrId, -- Destination
                    ospfPacketType,
                    ospfLsdbType,
                    ospfLsdbLsid,
                    ospfLsdbRouterId }
STATUS              current
DESCRIPTION
      "An ospfTxRetransmit trap signifies that an OSPF packet has been retransmitted on a non-
      virtual interface. All packets that may be re-transmitted are associated with an LSDB entry.
      The LS type, LS ID, and Router ID are used to identify the LSDB entry."
::= { ospfTraps 10 }

ospfVirtIfTxRetransmit NOTIFICATION-TYPE
   OBJECTS          {
                    ospfRouterId, -- The originator of the trap
                    ospfVirtIfAreaId,
                    ospfVirtIfNeighbor,
                    ospfPacketType,
                    ospfLsdbType,
                    ospfLsdbLsid,
                    ospfLsdbRouterId }
   STATUS           current
   DESCRIPTION
      "An ospfTxRetransmit trap signifies that an OSPF packet has been retransmitted on a virtual
      interface. All packets that may be retransmitted are associated with an LSDB entry. The LS
      type, LS ID, and Router ID are used to identify the LSDB entry."
::= { ospfTraps 11 }

ospfOriginateLsa NOTIFICATION-TYPE
   OBJECTS          {
                    ospfRouterId, -- The originator of the trap
                    ospfLsdbAreaId, -- 0.0.0.0 for AS Externals
                    ospfLsdbType,
                    ospfLsdbLsid,
                    ospfLsdbRouterId }
   STATUS           current
   DESCRIPTION
      "An ospfOriginateLsa trap signifies that a new LSA has been originated by this router. This
      trap should not be invoked for simple refreshes of LSAs (which happens every 30 minutes), but
      instead will only be invoked when an LSA is (re)originated due to a topology change. Addi-
      tionally, this trap does not include LSAs that are being flushed because they have reached
      MaxAge."
::= { ospfTraps 12 }

ospfMaxAgeLsa NOTIFICATION-TYPE
   OBJECTS          {ospfRouterId, -- The originator of the trap
                    ospfLsdbAreaId, -- 0.0.0.0 for AS Externals
                    ospfLsdbType,
                    ospfLsdbLsid,
                    ospfLsdbRouterId }
   STATUS           current
   DESCRIPTION
      "An ospfMaxAgeLsa trap signifies that one of the LSAs in the router's link-state database has
      aged to MaxAge."
::= { ospfTraps 13 }
```

```
ospfLsdbOverflow NOTIFICATION-TYPE
OBJECTS          {
      ospfRouterId, -- The originator of the trap
      ospfExtLsdbLimit }
   STATUS             current
   DESCRIPTION
      "An ospfLsdbOverflow trap signifies that the number of LSAs in the router's link-state data-
      base has exceeded ospfExtLsdbLimit."
::= { ospfTraps 14 }

ospfLsdbApproachingOverflow NOTIFICATION-TYPE
   OBJECTS          {\
        ospfRouterId, -- The originator of the trap
        ospfExtLsdbLimit }
   STATUS             current
   DESCRIPTION
      "An ospfLsdbApproachingOverflow trap signifies that the number of LSAs in the router's link-
      state database has exceeded ninety percent of ospfExtLsdbLimit."
   ::= { ospfTraps 15 }
```

SNMP Version 2 Ping Traps MIB

The following descriptions for the SNMPv2 ping traps are from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*:

```
pingProbeFailed          NOTIFICATION-TYPE
   OBJECTS {
      pingCtlTargetAddressType,
      pingCtlTargetAddress,
      pingResultsOperStatus,
      pingResultsIpTargetAddressType,
      pingResultsIpTargetAddress,
      pingResultsMinRtt,
      pingResultsMaxRtt,
      pingResultsAverageRtt,
      pingResultsProbeResponses,
      pingResultsSentProbes,
      pingResultsRttSumOfSquares,
      pingResultsLastGoodProbe
      }
   STATUS                current
   DESCRIPTION
      "Generated when a probe failure is detected when the
      corresponding pingCtlTrapGeneration object is set to
      probeFailure(0) subject to the value of
      pingCtlTrapProbeFailureFilter.  The object
      pingCtlTrapProbeFailureFilter can be used to specify the
      number of successive probe failures that are required
      before this notification can be generated."
::= { pingNotifications 1 }
```

```
pingTestFailed              NOTIFICATION-TYPE
   OBJECTS {
      pingCtlTargetAddressType,
      pingCtlTargetAddress,
      pingResultsOperStatus,
      pingResultsIpTargetAddressType,
      pingResultsIpTargetAddress,
      pingResultsMinRtt,
      pingResultsMaxRtt,
      pingResultsAverageRtt,
      pingResultsProbeResponses,
      pingResultsSentProbes,
      pingResultsRttSumOfSquares,
      pingResultsLastGoodProbe
      }
   STATUS                  current
   DESCRIPTION
      "Generated when a ping test is determined to have failed
      when the corresponding pingCtlTrapGeneration object is
      set to testFailure(1).  In this instance
      pingCtlTrapTestFailureFilter should specify the number of
      probes in a test required to have failed in order to
      consider the test as failed."
::= { pingNotifications 2 }

pingTestCompleted      NOTIFICATION-TYPE
   OBJECTS {
      pingCtlTargetAddressType,
      pingCtlTargetAddress,
      pingResultsOperStatus,
      pingResultsIpTargetAddressType,
      pingResultsIpTargetAddress,
      pingResultsMinRtt,
      pingResultsMaxRtt,
      pingResultsAverageRtt,
      pingResultsProbeResponses,
      pingResultsSentProbes,
      pingResultsRttSumOfSquares,
      pingResultsLastGoodProbe
      }
   STATUS                  current
   DESCRIPTION
      "Generated at the completion of a ping test when the
      corresponding pingCtlTrapGeneration object is set to
      testCompletion(4)."
::= { pingNotifications 3 }
```

SNMP Version 2 Traceroute Traps MIB

The following descriptions for the SNMPv2 traceroute traps are from RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*:

```
traceRoutePathChange        NOTIFICATION-TYPE
   OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
      }
   STATUS                   current
   DESCRIPTION
      "The path to a target has changed."
::= { traceRouteNotifications 1 }

traceRouteTestFailed        NOTIFICATION-TYPE
OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
      }
   STATUS                   current
   DESCRIPTION
      "Could not determine the path to a target."
::= { traceRouteNotifications 2 }

traceRouteTestCompleted     NOTIFICATION-TYPE
   OBJECTS {
      traceRouteCtlTargetAddressType,
      traceRouteCtlTargetAddress,
      traceRouteResultsIpTgtAddrType,
      traceRouteResultsIpTgtAddr
      }
   STATUS                   current
   DESCRIPTION
      "The path to a target has just been determined."
::= { traceRouteNotifications 3 }
```

SNMP Version 2 VRRP Traps MIB

The following descriptions for the SNMPv2 VRRP traps are from RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*:

```
--- vrrp trap definitions

   vrrpTrapPacketSrc          OBJECT-TYPE
      SYNTAX                  IpAddress
      MAX-ACCESS              accessible-for-notify
      STATUS                  current
      DESCRIPTION
         "The IP address of an inbound VRRP packet. Used by
         vrrpTrapAuthFailure trap."
   ::= { vrrpOperations 5 }

   vrrpTrapAuthErrorType      OBJECT-TYPE
      SYNTAX                  INTEGER {
                              invalidAuthType (1),
                              authTypeMismatch (2),
                              authFailure (3)
                              }
      MAX-ACCESS              accessible-for-notify
      STATUS                  current
      DESCRIPTION
         "Potential types of configuration conflicts.
         Used by vrrpAuthFailure trap."
```

Chapter 9

Summary of SNMP Configuration Statements

The following sections explain each of the SNMP configuration statements. The statements are organized alphabetically.

access

Syntax

```
access {
    context context-name {
        description description;
        group group-name {
            model usm;
            security-level (none | authentication | privacy);
            read-view view-name;
            write-view view-name;
        }
    }
    group group-name {
        user [ user-names ];
        model usm;
    }
    user [user-name] {
        authentication-type (none | md5 | sha);
        authentication-password authentication-password;
        privacy-password privacy-password
        privacy-type (none | des);
        clients {
            address restrict;
        }
    }
}
```

Hierarchy Level [edit snmp]

Description Specifies the SNMPv3 access level by context, group, and user.

Options The remaining statements are explained separately.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

agent-address

Syntax agent-address outgoing-interface;

Hierarchy Level [edit snmp trap-options]

Description Set the agent address of all SNMPv1 traps generated by this router. Currently, the only option is outgoing-interface, which sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.

Options outgoing-interface—Value of agent address of all SNMPv1 traps generated by this router. The outgoing-interface option sets the agent address of each SNMPv1 trap to the address of the outgoing interface of that trap.

Default: disabled (agent address is not specified in SNMPv1 traps)

Usage Guidelines See “Configure the Agent Address for SNMP Traps” on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

authentication-password

Syntax authentication-type *authentication-password*;

Hierarchy Level [edit snmp access user *user-name*]

Description Password used for authentication.

Options *authentication-password*—Contents of password used for authentication.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

authentication-type

Syntax authentication-type (none | md5 | sha);

Hierarchy Level [edit snmp access user *user-name*]

Description The type of authentication algorithm.

Options Includes the following authentication types:

- none—No security. SNMPv3 provides no authentication and no encryption on any SNMP information.
- md5—Message Digest Algorithm (see RFC 1321)
- sha—Secure Hash Algorithm (see NIST FIPS 180-1)

Usage Guidelines See "Configure SNMPv3 Access" on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

authorization

Syntax authorization *authorization*;

Hierarchy Level [edit snmp community *community-name*]

Description Set the access authorization for SNMP Get, GetBulk, GetNext, and Set requests.

Options *authorization*—Access authorization level:

- read-only—Enable Get, GetNext, and GetBulk requests.
- read-write—Enable all requests, including Set requests. You must configure a view to enable Set requests.

Default: read-only

Usage Guidelines See "Configure the SNMP Community String" on page 21.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

categories

Syntax categories [*categories*];

Hierarchy Level [edit snmp trap-group *group-name*]

Description Define the types of traps that will be sent to the targets of the named trap group.

Default If you omit the categories statement, all trap types are included in trap notifications.

Options *categories*—One or more trap types.
Values: authentication, chassis, configuration, link, remote-operations, rmon-alarm, routing, sonet-alarms, startup, vrrp-events

Usage Guidelines See "Configure SNMP Trap Groups" on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

clients

clients (for associating clients with communities)

Syntax
```
clients {
    address restrict;
}
```

Hierarchy Level [edit snmp community *community-name*]

Description Specify the IPv4 or IPv6 addresses of the SNMP client hosts that are authorized to use this community.

Default If you omit the clients statement, all SNMP clients using this community string are authorized to access the router.

Options *address*—Address of an SNMP client that is authorized to access this router. You must specify an address, not a hostname. To specify more than one client, include multiple *address* options.

restrict—(Optional) Do not allow the specified SNMP client to access the router.
Default: If you omit the restrict option after the address, access is permitted for this particular client.

Usage Guidelines See "Configure the SNMP Community String" on page 21.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

clients (for associating clients with an SNMPv3 user)

Syntax
```
clients {
    address restrict;
}
```

Hierarchy Level [edit snmp user *user-name*]

Description List of source address prefix ranges to accept.

Options *address*—IPv4 or IPv6 address of an SNMP client that is authorized to access this router. You must specify an address, not a hostname. To specify more than one client, include multiple *address* options.

restrict—(Optional) Do not allow the specified SNMP client to access the router.
Default: If you omit the restrict option after the address, access is permitted for this particular client.

Usage Guidelines See “Configure the SNMP Community String” on page 21.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

community

Syntax
```
community community-name {
    authorization authorization;
    clients {
        address restrict;
    }
    view view-name;
}
```

Hierarchy Level [edit snmp]

Description Define an SNMP community. An SNMP community authorizes SNMP clients based on source IP address of incoming SNMP request packets. A community also defines which MIB objects are available and the operations (read-only or read-write) allowed on those objects.

The SNMP client application specifies an SNMP community name in Get, GetBulk, GetNext, and Set SNMP requests.

Default If you omit the community statement, all SNMP requests are denied.

Options *community-name*—Community string. If the name includes spaces, enclose it in quotation marks (" ").

The remaining statements are explained separately.

Usage Guidelines See “Configure the SNMP Community String” on page 21.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

contact

Syntax contact *contact*;

Hierarchy Level [edit snmp]

Description Define the value of the MIB II sysContact object, which is the contact person for the managed system.

Options *contact*—Name of contact person. If the name includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See “Configure the System Contact” on page 19.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

context

Syntax
```
context context-name {
    description description;
    group group-name {
        model usm;
        security-level [none | authentication | privacy];
        read-view view-name;
        write-view view-name;
}
```

Hierarchy Level [edit snmp access]

Description A collection of management information accessible by an SNMP entity. An item of management information can exist in more than one context. An SNMP entity can have access to many contexts.

Options *context-name*—Sets a collection of management information accessible by an SNMP entity.

The remaining statements are explained separately.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

description

description (for describing the MIB II sysDescription object)

Syntax description *description*;

Hierarchy Level [edit snmp]

Description Define the value of the MIB II sysDescription object, which is the description of the system being managed.

Options *description*—System description. If the name includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See “Configure the System Description” on page 20.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

description (for describing the SNMPv3 context)

Syntax description *description*;

Hierarchy Level [edit snmp access context *context-name*]

Description Define the value of the context name accessible by the SNMP entity.

Options *description*—Describes the value of the context name. If the name includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

destination-port

Syntax destination-port <*port-number*>;

Hierarchy Level [edit snmp trap-group]

Description Assign a trap port number other than the default.

Options *port-number*—(Optional) SNMP trap port number.

Default: port number 162.

Usage Guidelines See “Configure SNMP Trap Groups” on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

engine-id

Syntax engine-id {
    local *engine-id*;
}

Hierarchy Level [edit snmp]

Description Set the SNMPv3 engine ID.

Options *engine-id*—An SNMPv3 engine's administratively unique identifier. It is used for identification, not for addressing. You must configure the local engine ID explicitly. The engine ID is in text format with its fifth octet equal to 4. If the engine ID is not configured, the system default IP address of the router is used as the default engine ID. The fifth octet of the default engine ID is 1.

Default: IPv4 address format with the fifth octet equal to 1.

Usage Guidelines See "Configure the Local Engine ID" on page 29.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

group

group (for associating a group with an SNMPv3 context)

Syntax group *group-name*;

Hierarchy Level [edit snmp access context *context-name*]

Description Associates a group with an SNMPv3 context.

Options *group-name*—SNMPv3 USM group name associated with an SNMPv3 context.

Usage Guidelines See "Configure SNMPv3 Access" on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

group (for creating an SNMPv3 group)

Syntax
```
group group-name {
    user [ user-names ];
    model usm;
}
```

Hierarchy Level [edit snmp access]

Description Creates an SNMPv3 group.

Options *group-name*—SNMPv3 group name created for an SNMPv3 group.

The remaining statements are described separately.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

interface

Syntax interface [*interface-names*];

Hierarchy Level [edit snmp]

Description Configure the interfaces on which SNMP requests can be accepted.

Default If you omit this statement, SNMP requests entering the router through any interface will be accepted.

Options *interface-names*—Names of one or more logical interfaces.

Usage Guidelines See “Configure the Interfaces on Which SNMP Requests Can Be Accepted” on page 28.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

location

Syntax location *location*;

Hierarchy Level [edit snmp]

Description Define the value of the MIB II sysLocation object, which is the physical location of the managed system.

Options *location*—Location of local system. You must enclose the name within quotation marks (" ").

Usage Guidelines See “Configure the System Location” on page 19.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

model

Syntax model usm;

Hierarchy Level [edit snmp access context *context-name*];
[edit snmp access group *group-name*]

Description Describes the security model used for SNMPv3 access.

Options usm—User-based Security Model (USM), which provides data integrity, data origin authentication, message replay protection, and protection against disclosure of the message payload.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

name

Syntax name *name*;

Hierarchy Level [edit snmp]

Description Set the system name from the command-line interface.

Options *name*—System name override.

Usage Guidelines See “Configure the System Name” on page 20.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

oid

Syntax oid *object-identifier* (include | exclude);

Hierarchy Level [edit snmp view *view-name*]

Description Specify an object identifier (OID) used to represent a subtree of MIB objects.

Options *object-identifier*—OID used to represent a subtree of MIB objects. All MIB objects represented by this statement have the specified OID as a prefix. It can be specified either by a sequence of dotted integers or by a subtree name.

include—Include the subtree of MIB objects represented by the specified OID.

exclude—Exclude the subtree of MIB objects represented by the specified OID.

Usage Guidelines See “Configure MIB Views” on page 28.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

privacy-password

Syntax privacy-password *privacy-password*;

Hierarchy Level [edit snmp access user *user-name*]

Description Password used for encryption.

Options *privacy-password*—Contents of password used for encryption.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

privacy-type

Syntax privacy-type (none | des);

Hierarchy Level [edit snmp access user *user-name*]

Description Level of privacy a user has with SNMPv3.

Options Includes the following privacy types:

- none—No security. SNMPv3 provides no authentication and no encryption on any SNMP information.
- des—Data Encryption Algorithm (see FIPS Publication 46-1).

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

read-view

Syntax read-view *view-name*;

Hierarchy Level [edit snmp access context *context-name* group *group-name*]

Description Specify read access for an SNMP user group.

Options *view-name*—The name of the view to which the SNMP user group has access.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

security-level

Syntax security-level (none | authentication | privacy);

Hierarchy Level [edit snmp access context *context-name* group *group-name*]

Description Level of security assigned to an SNMPv3 context.

Options includes three security levels:

- none—No security. SNMPv3 provides no authentication and no encryption on any SNMP information.
- authentication only—Provides authentication capability but no encryption on any SNMP information.
- privacy—Provides authentication and encryption on all SNMP information.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

snmp

Syntax snmp { ... }

Hierarchy Level [edit]

Description Configure SNMP.

Usage Guidelines See “Configure SNMP” on page 17.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

source-address

Syntax source-address *address*;

Hierarchy Level [edit snmp trap-options]

Description Set the source address of every SNMP trap packet sent by this router to a single address regardless of the outgoing interface. If the source address is not specified, the default is to use the address of the outgoing interface as the source address. Currently, the only value that can be specified for source address is lo0. The value lo0 indicates the source address of all SNMP trap packets will be set to the lowest loopback address configured at the interface lo0.

Options *address*—Source address of SNMP traps. Currently, the only value that can be specified is lo0.

Default: disabled (The source address is the address of outgoing interface)

Usage Guidelines See “Configure the Source Address for SNMP Traps” on page 24.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

targets

Syntax
```
targets {
    address;
}
```

Hierarchy Level [edit snmp trap-group *group-name*]

Description Configure one or more systems to receive SNMP traps.

Options *address*—IPv4 or IPv6 address of the system to receive traps. You must specify an address, not a hostname.

Usage Guidelines See “Configure SNMP Trap Groups” on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

traceoptions

Syntax

```
traceoptions {
    file size size files number;
    flag flag;
}
```

Hierarchy Level [edit snmp]

Description The output of the tracing operations is placed into log files in the /var/log directory. Each of these log files is named after the SNMP agent that generates it. Currently, the following logs are created in the /var/log directory when the traceoptions statement is used:

- chassisd
- craftd
- ilmid
- mib2d
- rmopd
- serviced
- snmpd

Options file *number*—(Optional) Maximum number of trace files per SNMP subagent. When a trace file (for example, snmpd) reaches its maximum size, it is archived by being renamed to snmpd.0. The previous snmpd.1 is renamed to snmpd.2, and so on. The oldest archived file is deleted.

Range: 2 through 1000 files
Default: 10 files

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements:

- all—Trace all SNMP events
- general—Trace general events
- interface-stats—Trace physical and logical interface statistics
- pdu—Trace SNMP request and response packets
- protocol-timeouts—Trace SNMP response timeouts
- routing-socket—Trace routing socket calls
- subagent—Trace subagent restarts
- timer—Trace internal timer events
- varbind-error—Trace variable binding errors

size *size*—(Optional) Maximum size in kilobytes (KB) of each trace file before it is closed and archived.

Range: 1 KB through the maximum file size supported on your system
Default: 1000 KB

Usage Guidelines See "Trace SNMP Activity" on page 32.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

trap-group

Syntax

```
trap-group group-name {
    categories [ categories ];
    destination-port <port-number>;
    targets {
        address;
    }
    version (all | v1 | v2);
}
```

Hierarchy Level [edit snmp]

Description Create a named group of hosts to receive the specified trap notifications. The name of the trap group is embedded in SNMP trap notification packets as one variable binding (varbind) known as the community name. At least one trap group must be configured for SNMP traps to be sent.

Options *group-name*—Name of the trap group. If the name includes spaces, enclose it in quotation marks (" ").

The remaining statements are explained separately.

Usage Guidelines See "Configure SNMP Trap Groups" on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

trap-options

Syntax
```
trap-options {
    agent-address outgoing-interface;
    source-address address;
}
```

Hierarchy Level [edit snmp]

Description Using SNMP trap options, you can set the source address of every SNMP trap packet sent by the router to a single address, regardless of the outgoing interface. In addition, you can set the agent address of each SNMPv1 trap. For more information on the contents of SNMPv1 traps, see RFC 1157.

Options The remaining statements are explained separately.

Default: disabled

Usage Guidelines See “Configure SNMP Trap Options” on page 23.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

user

user (for associating a list of users with an SNMPv3 group)

Syntax user [*user-name*];

Hierarchy Level [edit snmp access group *group-name*]

Description Specify a list of users associated with an SNMPv3 group.

Options *user-name*—SNMPv3 USM user name.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

user (for creating an SNMPv3 user)

Syntax user *user-name*;

Hierarchy Level [edit snmp access]

Description Specify a user for whom management operations are performed and authorized.

Options *user-name*—SNMPv3 USM user name.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration

version

Syntax version (all | v1 | v2);

Hierarchy Level [edit snmp trap-group *group-name*]

Description Specify the version number of SNMP traps.

Options all—Send an SNMPv1 and SNMPv2 trap for every trap condition.

v1—Send SNMPv1 traps only.

v2—Send SNMPv2 traps only.

Default: all

Usage Guidelines See “Configure SNMP Trap Groups” on page 25.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

view

view (for configuring MIB views)

Syntax
```
view view-name {
    oid object-identifier (include | exclude);
}
```

Hierarchy Level [edit snmp]

Description Define a MIB view. A MIB view identifies a group of MIB objects. Each MIB object in a view has a common OID prefix. Each object identifier represents a subtree of the MIB object hierarchy. The view statement uses a view to specify a group of MIB objects on which to define access. To enable a view, you must associate the view with a community by including the view statement at the [edit snmp community *community-name*] hierarchy level.


Note

To remove an OID completely, use the delete view all oid oid-number command but omit the include parameter.

Options *view-name*—Name of the view

The remaining statements are explained separately.

Usage Guidelines See “Configure MIB Views” on page 28.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also community on page 99

view (for associating MIB views with a community)

Syntax view *view-name*;

Hierarchy Level [edit snmp community *community-name*]

Description Associate a view with a community. A view represents a group of MIB objects.

Options *view-name*—Name of the view. You must use a view name already configured in the view statement at the [edit snmp] hierarchy level.

Usage Guidelines See “Configure MIB Views” on page 28.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

write-view

Syntax write-view *view-name*;

Hierarchy Level [edit snmp access context *context-name* group *group-name*]

Description Specifies write-view access for an SNMP user group.

Options *view-name*—The name of the view to which the SNMP user group has access.

Usage Guidelines See “Configure SNMPv3 Access” on page 30.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

Part 3

RMON Alarms and Events

Chapter 10
Configure RMON Alarms and Events

The JUNOS software supports monitoring routers from remote devices. These values are measured against thresholds and trigger events when the thresholds are crossed. You configure RMON alarm and event entries to monitor the value of a MIB object.

For more information on configuring RMON alarm and event entries, see “Configure RMON Alarms and Events” on page 115 and “Summary of RMON Alarm and Event Configuration Statements” on page 133.

For more information on monitoring integer-valued MIB objects, see “Monitor RMON Alarms and Events” on page 123.

To configure RMON alarm and event entries, you include statements at the [edit snmp] hierarchy level of the configuration.

```
[edit snmp]
rmon {
    alarm index {
        description text-description;
        falling-event-index index;
        falling-threshold integer;
        interval seconds;
        rising-event-index index;
        rising-threshold integer;
        sample-type (absolute-value | delta-value);
        startup-alarm (falling-alarm | rising alarm | rising-or-falling-alarm);
        variable oid-variable;
    event index {
        community community-name;
        description description;
        sample-type type;
    }
}
```

This chapter describes the minimum required configuration and discusses the following tasks for configuring RMON:

- Minimum RMON Alarm and Event Entry Configuration on page 116
- Configure an Alarm Entry and Its Attributes on page 116
- Configure an Event Entry and Its Attributes on page 120
- Example: Configure an RMON Alarm and Event Entry on page 121

Minimum RMON Alarm and Event Entry Configuration

To enable RMON on the router, you must configure an alarm entry and an event entry. To do this, include the following statements at the [edit snmp rmon] hierarchy level:

```
[edit snmp rmon]
alarm index {
    rising-event-index index;
    rising-threshold integer;
    sample-type type;
    variable oid-variable;
}
event index;
```

Configure an Alarm Entry and Its Attributes

An alarm entry monitors the value of a MIB variable. You can configure how often the value is sampled, the type of sampling to perform, and what event to trigger if a threshold is crossed.

This section discusses the following topics:

Configure the Alarm Entry

An alarm entry monitors the value of a MIB variable. The rising-event-index, rising-threshold, sample-type, and variable statements are mandatory. All other statements are optional.

To configure the alarm entry, include the alarm statement and specify an index at the [edit snmp rmon] hierarchy level.

```
[edit snmp rmon]
alarm index {
    description description;
    falling-event-index index;
    falling-threshold integer;
    interval seconds;
    rising-event-index index;
    rising-threshold integer;
    sample-type (absolut-value | delta-value);
    startup-alarm (falling-alarm | rising alarm | rising-or-falling-alarm);
    variable oid-variable;
}
```

index is an integer that identifies an alarm or event entry.

Configure the Description

The description is a text string that identifies the alarm or event entry.

To configure the description, include the description statement and a description of the alarm entry at the [edit snmp rmon alarm index] hierarchy level:

```
[edit snmp rmon alarm index]
description description;
```

Configure the Falling Event Index or Rising Event Index

The falling event index identifies the event entry that is triggered when a falling threshold is crossed. The rising event index identifies the event entry that is triggered when a rising threshold is crossed.

To configure the falling event index or rising event index, include the falling-event-index or rising-event-index statement and specify an index at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
falling-event-index index;
rising-event-index index;
```

index can be 0 through 65,535. The default for the both the falling and rising event index is 0.

Configure the Falling Threshold and Rising Threshold

The falling threshold is the lower threshold for the monitored variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup alarm is equal to falling-alarm or rising-or-falling-alarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising threshold. You must specify the falling threshold as an integer. Its default is 20 percent less than the rising threshold.

The rising threshold is the upper threshold for the monitored variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated startup-alarm is equal to rising-alarm or rising-or-falling-alarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling threshold. You must specify the rising threshold as an integer.

To configure the falling threshold or rising threshold, include the falling-threshold or rising-threshold statement at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
falling-threshold integer;
rising-threshold integer;
```

integer can be -2,147,483,647 to 2,147,483,647.

Configure the Interval

The interval represents the period of time, in seconds, over which the monitored variable is sampled and compared with the rising and falling thresholds.

To configure the interval, include the interval statement and specify the number of seconds at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
interval seconds;
```

seconds can be 1 through 2,147,483,647. The default is 60 seconds.

Configure the Sample Type

The sample type identifies the method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absolute-value, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is delta-value, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds.

To configure the sample type, include the sample-type statement and specify the type of sample at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
sample-type (absolute-value | delta-value);
```

- absolute-value—Actual value of the selected variable is compared against the thresholds.
- delta-value—Difference between samples of the selected variable is compared against the thresholds.

Configure the Startup Alarm

The startup alarm identifies the type of alarm that can be sent when this entry is first activated. You can specify it as falling-alarm, rising-alarm, or rising-or-falling-alarm.

To configure the startup alarm, include the startup-alarm statement and specify the type of alarm at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);
```

- falling-alarm—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.
- rising-alarm—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.
- rising-or-falling-alarm—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.

The default is rising-or-falling-alarm.

Configure the Variable

The variable identifies the the MIB object that is being monitored.

To configure the variable, include the variable statement and specify the object identifier or object name at the [edit snmp rmon alarm *index*] hierarchy level:

```
[edit snmp rmon alarm index]
variable oid-variable;
```

oid-variable is a dotted decimal (for example, .1.3.6.1.2.1.2.1.2.2.1.10.1) or MIB object name (for example, ifInOctets.1.).

Configure an Event Entry and Its Attributes

An event entry generates a notification for an alarm entry when its rising or falling threshold is crossed. You can configure the type of notification that is generated. To configure the event entry, include the event statement at the [edit snmp rmon] hierarchy level. All statements except the event statement are optional.

```
[edit snmp rmon]
event index {
    community community-name;
    description description;
    sample-type type;
}
```

The *index* variable of an event entry is an integer that identifies an entry event.

The *community-name* variable of an event entry is the trap group that is used when generating a trap. If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group. If nothing is configured, all the trap groups are examined, and traps are sent using each group with the rmon-alarm category set..

The *description* variable of an event entry is a text string that identifies the entry.

The *type* variable of an event entry specifies where the event is to be logged. You can specify the type as one of the following:

- log—Adds the event entry to the logTable.
- log-and-trap—Sends an SNMP trap and creates a log entry.
- none—Sends no notification.
- snmptrap—Sends an SNMP trap.

The default for the event entry type is log-and-trap.

Example: Configure an RMON Alarm and Event Entry

Configure an RMON alarm and event entry:

```
[edit snmp]
   rmon {
      alarm 100 {
         description “input traffic on fxp0”;
         falling-event-index 100;
         falling-threshold 10000;
         interval 60;
         rising-event-index 100;
         rising-threshold 100000;
         sample-type delta-value;
         startup-alarm rising-or-falling-alarm;
         variable ifInOctets.1;
   }
      event 100 {
         community bedrock;
         description “emergency events”;
         sample-type log-and-trap;
      }
   }
}
```

Chapter 11

Monitor RMON Alarms and Events

The remote monitoring (RMON) alarms and events feature can be used to to monitor integer-valued MIB objects, standard or enterprise-specific, on a Juniper Networks router. Configuration and operational information are in the MIB objects defined in alarmTable, eventTable, and logTable in RFC 2819. Additional information is defined by the Juniper Networks enterprise-specific extension to alarmTable defined in jnxRmonMIB (jnx-rmon-mib.txt).

This chapter covers the following main topics:

RMON Alarms

An RMON alarm identifies:

- A specific MIB object that is monitored.
- The frequency at which it is sampled.
- The method of sampling.
- The thresholds against which the monitored values are compared.

An RMON alarm can also identify a specific eventTable entry to be triggered when a threshold is crossed.

Configuration and operational values are defined in alarmTable in RFC 2819. Additional operational values are defined in Juniper Networks enterprise-specific extensions to alarmTable (jnxRmonAlarmTable).

This section covers the following topics:

alarmTable

alarmTable in the RMON MIB allows you to monitor and poll the following:

- alarmIndex—The index value for alarmTable that identifies a specific entry.
- alarmInterval—The interval in seconds over which data is sampled and compared with the rising and falling thresholds.
- alarmVariable—The MIB variable that is monitored by the alarm entry.
- alarmSampleType—The method of sampling the selected variable and calculating the value to be compared against the thresholds.
- alarmValue—The value of the variable during the last sampling period. This is the value that is compared with the rising and falling thresholds.
- alarmStartupAlarm—The alarm that can be sent when the entry is first activated.
- alarmRisingThreshold—The upper threshold for the sampled variable.
- alarmFallingThreshold—The lower threshold for the sampled variable.
- alarmRisingEventIndex—The eventTable entry that is used when a rising threshold is crossed.
- alarmFallingEventIndex—The eventTable entry that is used when a falling threshold is crossed.
- alarmStatus—Add and remove entries from the table. It can also be used to change the state of an entry to allow modifications.


Note

If this object is not set to valid, no action will be taken by the associated event alarm.

jnxRmonAlarmTable

The jnxRmonAlarmTable is a Juniper Networks enterprise-specific extension to alarmTable. It provides additional operational information and includes the following objects:

- jnxRmonAlarmGetFailCnt—The number of times the internal Get request for the variable monitored by this entry has failed.
- jnxRmonAlarmGetFailTime—The value of sysUpTime when an internal Get request for the variable monitored by this entry last failed.
- jnxRmonAlarmGetFailReason—The reason an internal Get request for the variable monitored by this entry last failed.
- jnxRmonAlarmGetOkTime—The value of sysUpTime when an internal Get request for the variable monitored by this entry succeeded and the entry left the getFailure state.
- jnxRmonAlarmState—The current state of this RMON alarm entry.

To view the Juniper Networks enterprise-specific extensions to the RMON alarm and event MIB, see www.juniper.net/techpubs/software/junos56/swconfig56-net-mgmt/html/mib-jnx-rmon.txt. For more information on the Juniper Networks enterprise-specific extensions to the RMON events and alarms MIB, see “Interpret the RMON Events and Alarms MIB” on page 219.

Use alarmTable to Monitor MIB Objects

To use alarmTable to monitor a MIB object, perform the following tasks:

- Create an Alarm Entry on page 125
- Configure the Alarm MIB Objects on page 126
- Activate a New Row in alarmTable on page 128
- Modify an Active Row in alarmTable on page 129
- Deactivate a Row in alarmTable on page 129

Create an Alarm Entry

To create an alarm entry, first create a new row in alarmTable using the alarmStatus object. For example, create alarm #1 using the UCD command-line utilities:

```
snmpset -Os -v2c router community alarmStatus.1 i createRequest
```

Configure the Alarm MIB Objects

Once you have created the new row in alarmTable, configure the following alarm MIB objects:

- alarmInterval on page 126
- alarmVariable on page 126
- alarmSampleType on page 127
- alarmValue on page 127
- alarmStartupAlarm on page 127
- alarmRisingThreshold on page 127
- alarmFallingThreshold on page 128
- alarmOwner on page 128
- alarmRisingEventIndex on page 128
- alarmFallingEventIndex on page 128


Note

Other than alarmStatus, none of the objects in the entry can be modified if the associated alarmStatus object is set to valid.

alarmInterval

The interval in seconds over which data is sampled and compared with the rising and falling thresholds. For example, to set alarmInterval for alarm #1 to 30 seconds, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmInterval.1 i 30
```

alarmVariable

The object identifier of the variable to be sampled. During a Set request, if the supplied variable name is not available in the selected MIB view, a badValue error is returned. If at any time the variable name of an established alarmEntry is no longer available in the selected MIB view, the probe changes the status of alarmVariable to invalid. For example, to identify ifInOctets.61 as the variable to be monitored, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmVariable.1 o .1.3.6.1.2.1.2.2.1.10.61
```

alarmSampleType

The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absoluteValue, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is deltaValue, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds. For example, to set alarmSampleType for alarm #1 to deltaValue, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmSampleType.1 i deltaValue
```

alarmValue

The value of the variable during the last sampling period. This is the value that is compared with the rising and falling thresholds. If the sample type is deltaValue, this value equals the difference between the samples at the beginning and end of the period. If the sample type is absoluteValue, this value equals the sampled value at the end of the period.

alarmStartupAlarm

An alarm that is sent when this entry is first set to valid. If the first sample after this entry becomes valid is greater than or equal to risingThreshold, and alarmStartupAlarm is equal to risingAlarm or or risingOrFallingAlarm, then a single rising alarm is generated. If the first sample after this entry becomes valid is less than or equal to fallingThreshold and alarmStartupAlarm is equal to fallingAlarm or risingOrFallingAlarm, then a single falling alarm is generated. For example, to set alarmStartupAlarm for alarm #1 to risingOrFallingAlarm, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmStartupAlarm.1 i risingOrFallingAlarm
```

alarmRisingThreshold

A threshold for the sampled variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated alarmStartupAlarm is equal to risingAlarm or risingOrFallingAlarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches alarmFallingThreshold. For example, to set alarmRisingThreshold for alarm #1 to 100000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingThreshold.1 i 100000
```

alarmFallingThreshold

A threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated alarmStartupAlarm is equal to fallingAlarm or risingOrFallingAlarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches alarmRisingThreshold. For example, to set alarmFallingThreshold for alarm #1 to 10000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 10000
```

alarmOwner

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

alarmRisingEventIndex

The index of the eventEntry object that is used when a rising threshold is crossed. If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmRisingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingEventIndex.1 i 10
```

alarmFallingEventIndex

The index of the eventEntry object that is used when a falling threshold is crossed.If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmFallingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingEventIndex.1 i 10
```

Activate a New Row in alarmTable

To activate a new row in alarmTable, set alarmStatus to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

Modify an Active Row in alarmTable

To modify an active row, first set alarmStatus object to underCreation using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i underCreation
```

Then change the row contents using an SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 1000
```

Finally, activate the row by setting alarmStatus object to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

Deactivate a Row in alarmTable

To deactivate a row in alarmTable, set alarmStatus to invalid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i invalid
```

RMON Events

An RMON event allows you to log the crossing of thresholds of other MIB objects. It is defined in eventTable for the RMON MIB.

This section covers the following topics:

eventTable

eventTable contains the following objects:

- eventIndex—An index that uniquely identifies an entry in eventTable. Each entry defines one event that is to be generated when the appropriate conditions occur.
- eventDescription—A comment describing the event entry.
- eventType—Type of notification that the probe makes about this event.
- eventCommunity—Trap group used if an SNMP trap is to be sent. If eventCommunity is not configured, a trap is sent to each trap group configured with the rmon-alarm category.
- eventLastTimeSent—Value of sysUpTime when this event entry last generated an event.

- eventOwner—Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.
- eventStatus—Status of this event entry.

If this object is not set to valid, no action is taken by the associated event entry. When this object is set to valid, all previous log entries associated with this entry (if any) will be deleted.

Use eventTable to Log Alarms

To use eventTable to log alarms, perform the following tasks:

- Create an Event Entry on page 130
- Configure the MIB Objects on page 130
- Activate the New Row in eventTable on page 132
- Deactivate a Row in eventTable on page 132

Create an Event Entry

The RMON eventTable controls the generation of notifications from the router. Notifications can be logs (entries to logTable and syslogs) or SNMP traps. Each event entry can be configured to generate any combination of these notifications (or no notification). When an event specifies that an SNMP trap is to be generated, the trap group that is used when sending the trap is specified by the value of the associated eventCommunity object. Consequently, the community in the trap message will match the value specified by eventCommunity. If nothing is configured for eventCommunity, a trap is sent using each trap group that has the rmon-alarm category configured.

Configure the MIB Objects

Once you have created the new row in eventTable, set the following objects:

- eventType on page 131
- eventCommunity on page 131
- eventOwner on page 131
- eventDescription on page 132

The eventType object is required. All other objects are optional.

eventType

The type of notification that the router generates when the event is triggered.

This object can be set to the following values:

- log—Adds the event entry to logTable.
- log-and-trap—Sends an SNMP trap and creates a log entry.
- none—Sends no notification.
- snmptrap—Sends an SNMP trap.

For example, to set eventType for event #1 to log-and-trap, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventType.1 i log-and-trap
```

eventCommunity

The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set. For example, to set eventCommunity for event #1 to boy-elroy, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventCommunity.1 s "boy-elroy"
```

The eventCommunity object is optional. If you do not set this object, then the field is left blank.

eventOwner

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

For example, to set eventOwner for event #1 to george jetson, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventOwner.1 s "george jetson"
```

eventOwner object is optional. If you do not set this object, then the field is left blank.

eventDescription

Any text string specified by the creating management application or the CLI. The use of this string is application dependent.

For example, to set eventDescription for event #1 to spacelys sprockets, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventDescription.1 s "spacelys sprockets"
```


Note

The eventDescription object is optional. If you do not set this object, then the field is left blank.

Activate the New Row in eventTable

To activate the new row in eventTable, set eventStatus to valid using an SNMP Set request such as:

```
snmpset -Os -v2c router community eventStatus.1 i valid
```

Deactivate a Row in eventTable

To deactivate a row in eventTable, set eventStatus to invalid using an SNMP Set request such as:

```
snmpset -Os -v2c router community eventStatus.1 i invalid
```

Chapter 12

Summary of RMON Alarm and Event Configuration Statements

The following sections explain each of the RMON alarm and event configuration statements. The statements are organized alphabetically.

alarm

Syntax

```
alarm index {
    description text-description;
    falling-event-index index;
    falling-threshold integer;
    interval seconds;
    rising-event-index index;
    rising-threshold integer;
    sample-type (absolute-value | delta-value);
    startup-alarm (falling-alarm | rising-alarm | rising-or-falling alarm);
    variable oid-variable;
}
```

Hierarchy Level [edit snmp rmon]

Description Configure RMON alarm entries.

Options *index*—Identifies this alarm entry as an integer.

The remaining statements are explained separately.

Usage Guidelines See “Configure an Alarm Entry and Its Attributes” on page 116.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also event on page 135.

community

Syntax community *community-name*;

Hierarchy Level [edit snmp rmon event *index*]

Description The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set.

Options *community-name*—Identifies the trap group that is used when generating a trap if the event is configured to send traps.

Usage Guidelines See “Configure an Event Entry and Its Attributes” on page 120.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

description

Syntax description *description*;

Hierarchy Level [edit snmp rmon alarm *index*],
[edit snmp rmon event *index*]

Description Text description of alarm or event.

Options *description*—Text description of an alarm or event entry. If the description includes spaces, enclose it in quotation marks (" ").

Usage Guidelines See “Configure the Description” on page 117 and “Configure an Event Entry and Its Attributes” on page 120.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

event

Syntax

```
event index {
    community community-name;
    description text-description;
    type type;
}
```

Hierarchy Level [edit snmp rmon]

Description Configure RMON event entries.

Options *index*—Identifies a specific event entry.

The remaining statements are explained separately.

Usage Guidelines See “Configure an Event Entry and Its Attributes” on page 120.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also alarm on page 133.

falling-event-index

Syntax falling-event-index *index*;

Hierarchy Level [edit snmp rmon alarm *index*]

Description The index of the event entry that is used when a falling threshold is crossed. If this value is zero, no event is triggered.

Options *index*—Index of the event entry that is used when a falling threshold is crossed.
Range: 0 through 65,535
Default: 0 seconds

Usage Guidelines See “Configure the Falling Event Index or Rising Event Index” on page 117.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also rising-event-index on page 137

falling-threshold

Syntax falling-threshold *integer*;

Hierarchy Level [edit snmp rmon alarm *integer*]

Description The lower threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated startup-alarm is equal to falling-alarm or rising-or-falling-alarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches the rising-threshold.

Options *integer*—The lower threshold for the alarm entry.
Range: -2,147,483,648 through 2,147,483,647
Default: 20 percent less than rising-threshold

Usage Guidelines See "Configure the Falling Threshold and Rising Threshold" on page 118.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also rising-threshold on page 137.

interval

Syntax interval *seconds*;

Hierarchy Level [edit snmp rmon alarm *index*]

Description Interval between samples.

Options *interval*—Time between samples, in seconds.
Range: 1 through 2,147,483,647 seconds
Default: 60 seconds

Usage Guidelines See "Configure the Interval" on page 118.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

rising-event-index

Syntax rising-event-index *index*;

Hierarchy Level [edit snmp rmon alarm *index*]

Description The index of the event entry that is used when a rising threshold is crossed. If this value is zero, no event is triggered.

Options *index*—Index of the event entry that is used when a rising threshold is crossed.
Range: 0 through 65,535
Default: 0

Usage Guidelines See “Configure the Falling Event Index or Rising Event Index” on page 117.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also falling-event-index on page 135

rising-threshold

Syntax rising-threshold *integer*;

Hierarchy Level [edit snmp rmon alarm *index*]

Description The upper threshold for the sampled variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated startup-alarm is equal to falling-alarm or rising-or-falling-alarm. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches the falling-threshold.

Options *integer*—The lower threshold for the alarm entry.
Range: -2,147,483,648 through 2,147,483,647
Default: 0

Usage Guidelines See “Configure the Falling Threshold and Rising Threshold” on page 118.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

See Also falling-threshold on page 136

rmon

Syntax rmon { ... }

Hierarchy Level [edit snmp]

Description Configure Remote Monitoring.

Usage Guidelines See “Configure RMON Alarms and Events” on page 115.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

sample-type

sample-type (for RMON Alarms)

Syntax sample-type (absolute-value | delta-value);

Hierarchy Level [edit snmp rmon alarm *index*]

Description sample-type—Method of sampling the selected variable.

Options

- absolute-value—Actual value of the selected variable is used when comparing against the thresholds.
- delta-value—Difference between samples of the selected variable is used when comparing against the thresholds.

Usage Guidelines See “Configure the Sample Type” on page 119.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

sample-type (for RMON Events)

Syntax sample-type *type*;

Hierarchy Level [edit snmp rmon event *index*]

Description Type of notification generated when a threshold is crossed.

Options *type*—Type of notification. It can be one of the following:

- log—Add an entry to logTable.
- log-and-trap—Send an SNMP trap and make a log entry.
- none—No notifications are sent.
- snmptrap—Send an SNMP trap.

Default: log-and-trap

Usage Guidelines See "Configure an Event Entry and Its Attributes" on page 120.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

startup-alarm

Syntax startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm);

Hierarchy Level [edit snmp rmon alarm *index*]

Description The alarm that can be sent upon entry startup.

Options (falling-alarm | rising-alarm | rising-or-falling-alarm) E\

- falling-alarm—Generated if the first sample after the alarm entry becomes active is less than or equal to the falling threshold.
- rising-alarm—Generated if the first sample after the alarm entry becomes active is greater than or equal to the rising threshold.
- rising-or-falling-alarm—Generated if the first sample after the alarm entry becomes active satisfies either of the corresponding thresholds.

Default: rising-or-falling-alarm

Usage Guidelines See "Configure the Startup Alarm" on page 119.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

variable

Syntax variable *oid-variable*;

Hierarchy Level [edit snmp rmon alarm *index*]

Description Object identifier (OID) of MIB variable to be monitored.

Options *oid-variable*—OID of the MIB variable that is being monitored. The OID can be a dotted decimal (for example, .1.3.6.1.2.1.2.1.2.2.1.10.1) or use the MIB objects name (for example, ifInOctets.1.).

Usage Guidelines See “Configure the Variable” on page 119.

Required Privilege Level snmp—To view this statement in the configuration.
snmp-control—To add this statement to the configuration.

Part 4

Interpret the Juniper Networks Enterprise-Specific MIBs

Chapter 13

Interpret the Chassis MIB

The enterprise-specific chassis MIB provides information on the router and its components. MIB objects represent each component and the status of the components. The chassis MIB has four branches:

- jnxProducts on page 143
- jnxServices on page 144
- jnxMIBs on page 144
- jnxTraps on page 209

You can retrieve information from the MIB using any network management system (NMS). For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-chassis.txt.

jnxProducts

The object identifier for the jnxProducts branch of the chassis MIB is {juniperMIB 1}. This branch of the MIB describes the Juniper Networks routers and their components, such as product line, product name, model, number of slots, and media space for holding PICs. It also provides information on the system's power supply state, board voltages, fans, temperatures, and air flow. In general, this branch of the chassis MIB is rarely polled for information because it is descriptive. However, you can poll this branch of the chassis MIB to determine the sysObjectId of a router as defined by MIB-II.

The last number in each sysObjectId, shown in Table 11, corresponds to the router model and therefore does not change.

Table 11: Router Models and Their sysObjectIds

Model	SysObjectID	Description
M40	1.3.6.1.4.1.2636.1.1.1.2.1	jnxProductNameM40
M20	1.3.6.1.4.1.2636.1.1.1.2.2	jnxProductNameM20
M160	1.3.6.1.4.1.2636.1.1.1.2.3	jnxProductNameM160
M10	1.3.6.1.4.1.2636.1.1.1.2.4	jnxProductNameM10
M5	1.3.6.1.4.1.2636.1.1.1.2.5	jnxProductNameM5
T640	1.3.6.1.4.1.2636.1.1.1.2.6	jnxProductNameT640
T320	1.3.6.1.4.1.2636.1.1.1.2.7	jnxProductNameT320
M40e	1.3.6.1.4.1.2636.1.1.1.2.8	jnxProductNameM40e

jnxServices

The object identifier for the jnxServices branch is {juniperMIB 2}. The jnxServices branch of the chassis MIB is a placeholder for future information.

jnxMIBs

The object identifier for the jnxMIBs branch is {juniperMIB 3} and includes one main subbranch, jnxBoxAnatomy, whose object identifier is {jnxMIBs 1}. Other than the chassis MIB, the other enterprise-specific MIBs are also branches of jnxMIBs. These enterprise-specific MIBs include:

- MPLS MIB—Whose object identifier is {jnxMIBs 2}.
- Juniper Networks enterprise-specific extensions to the Interface MIB—Whose object identifier is {jnxMIBs 3}.
- Alarm MIB—Whose object identifier is {jnxMIBs 4}.
- Firewalls MIB—Whose object identifier is {jnxMIBs 5}.
- Destination Class Usage MIB—Whose object identifier is {jnxMIBs 6}.
- Juniper Networks enterprise-specific extensions to the pingMIB—Whose object identifier is {jnxMIBs 7}.
- Juniper Networks enterprise-specific extensions to the traceroute MIB—Whose object identifier is {jnxMIBs 8}.
- ATM MIB—Whose object identifier is {jnxMIBs 10}.
- IPv6 and ICMPv6 MIB—Whose object identifier is {jnxMIBs 11}.
- IPv4 MIB—Whose object identifier is {jnxMIBs 12}.

- Juniper Networks enterprise-specific extensions to the RMONMIB—Whose object identifier is {jnxMIBs 13}.
- Juniper Networks enterprise-specific extensions to the LDP traps MIB—Whose object identifier is {jnxMIBs 14}.
- Class-of-Service MIB—Whose object identifier is {jnxMIBs 15}.
- Source Class Usage MIB—Whose object identifier is {jnxMIBs 16}.
- Reverse Path Forwarding MIB—Whose object identifier is {jnxMIBs 17}.
- Configuration Management MIB—Whose object identifier is {jnxMIBs 18}.

For more information on these MIBs, see "Juniper Networks Enterprise-Specific MIBs" on page 55.

jnxBoxAnatomy

The object identifier for the jnxBoxAnatomy MIB is {jnxMIBs 1}.

The jnxBoxAnatomy MIB has 13 sections. See the following topics for more information:

- jnxBoxClass—See "Top-Level Objects" on page 146.
- jnxBoxDescr—See "Top-Level Objects" on page 146.
- jnxBoxSerialNo—See "Top-Level Objects" on page 146.
- jnxBoxRevision—See "Top-Level Objects" on page 146.
- jnxBoxInstalled—See "Top-Level Objects" on page 146.
- jnxContainersTable—See jnxContainersTable on page 146.
- jnxContentsLastChange—See jnxContentsLastChange on page 152.
- jnxContentsTable—See jnxContentsTable on page 153.
- jnxLEDLastChange—See jnxLEDLastChange on page 160.
- jnxLEDTable—See jnxLEDTable on page 161.
- jnxFilledLastChange—See jnxFilledLastChange on page 164.
- jnxFilledTable—See jnxFilledTable on page 164.
- jnxOperatingTable—See jnxOperatingTable on page 172.
- jnxRedundancyTable—See jnxRedundancyTable on page 178.
- jnxFruTable—See jnxFruTable on page 183.

Top-Level Objects

The following branches of the jnxBoxAnatomy MIB are top-level objects:

- jnxBoxClass—The object identifier for the jnxBoxClass object is {jnxBoxAnatomy 1}. This object classifies the chassis product line.
- jnxBoxDescr—The object identifier for the jnxBoxDescr object is {jnxBoxAnatomy 2}. This object describes the chassis name and model.
- jnxBoxSerialNo—The object identifier for the jnxBoxSerialNo object is {jnxBoxAnatomy 3}. This object indicates the serial number of the chassis. jnxBoxSerialNo remains blank if the serial number is unknown or unavailable.
- jnxBoxRevision—The object identifier for the jnxBoxRevision object is {jnxBoxAnatomy 4}. This object indicates the last revision of the chassis.
- jnxBoxInstalled—The object identifier for the jnxBoxInstalled object is {jnxBoxAnatomy 5}. This object indicates the last time the box was installed and operational, represented by the sysUpTime value.

jnxContainersTable

The object identifier for the jnxContainersTable object is {jnxBoxAnatomy 6}. This object shows the structure of the chassis.

You can use the jnxContainersTable object to retrieve specific information on the router, such as how many of each component the router can contain. For example, the jnxContainersTable of an M20 router indicates that the router can accommodate four FPCs; however, it does not describe how many FPCs the router actually has.

For more information on how many FPCs are actually on a router, see jnxContentsTable on page 153.

Entries within the jnxContainersTable object are represented by the jnxContainersEntry object, whose object identifier is {jnxContainersTable 1}. This jnxContainersEntry contains the following objects, which describe the contents of a particular router:

- jnxContainersIndex—The index value of an entry in the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 1}, which corresponds with jnxContainersType and jnxContainersDescr.

- jnxContainersView—The orientation of a container from the front of the router, whose object identifier is {jnxContainersEntry 2}. This object also indicates that the container is embedded in the router and how it is accessible from corresponding views. The value of this object is a bitmap represented as a sum. If multiple bits are set, you can access the container from that set of views. The values represent the bit positions and their corresponding views as follows:

  - 1—Front
  - 2—Rear
  - 4—Top
  - 8—Bottom
  - 16—Left side
  - 32—Right side

  For each view plane, if specified counters are scattered in various views, the numbering sequence starts from left to right and then from top to bottom, as follows:

  - Left side
  - Right side
  - Top
  - Bottom
  - Front
  - Rear

References to left and right sides are based on the view from the front of the chassis.

In accordance with network management conventions, all indexes in the MIB begin with 1, not 0, although the slot number might be labeled 0.

- jnxContainersLevel—The abstraction level of the box or components for the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 3}. The level is enumerated from the outside to the inside, and from the outer layer to the inner layer.

  For example, if the top level (level 0) of the box refers to the chassis frame, then the next level (level 1) refers to the FPC slot within the chassis frame. Finally, the PIC space within the FPC slot of the chassis corresponds to level 2.

- jnxContainersWithin—The container housing the entry at the next-higher level of the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 4}.

  For example, the within value for jnxMediaCardSpacePIC.0 is 7. Because the jnxM20SlotFPC.0 retains an index value of 7, the FPC houses the PIC.

- jnxContainersType—The component of the chassis MIB at a specific index, view, level, and within value for the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 5}.

- jnxContainersDescr—The description of the component in the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 6}.

- jnxContainersCount—The maximum number of a given component that the router can accommodate within the jnxContainersEntry object, whose object identifier is {jnxContainersEntry 7}.

  For example, the M20 router can house a specific maximum number of FPCs within the chassis frame. The maximum number is not necessarily the actual number of FPCs; this can change dynamically.

Table 12 through Table 19 provide examples of jnxContainersEntry objects in the jnxContainersTable. The following column headings for each table are abbreviated to correspond to the parts of the jnxContainersEntry objects:

- Index—jnxContainersIndex
- View—jnxContainersView
- Level—jnxContainersLevel
- Within—jnxContainersWithin
- Type—jnxContainersType
- Description—jnxContainersDescr
- Count—jnxContainersCount

Table 12 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M40 router.

Table 12: jnxContainersEntry Objects in the jnxContainersTable of an M40 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM40.0	Chassis frame compartment	1
2	2	1	1	jnxSlotPowerSupply.0	Power supply compartment	2
3	3	1	1	jnxSlotCoolingImpeller.0	Impeller compartment	2
4	2	1	1	jnxSlotCoolingFan.0	Fan compartment	3
5	2	1	1	jnxSlotHostCtrl.0	Host controller compartment	1
6	1	1	1	jnxSlotSCB.0	SCB slot	1
7	1	1	1	jnxSlotFPC.0	FPC slot	8
8	1	2	7	jnxMediaSlotCardPIC.0	PIC space	4
9	2	1	1	jnxSlotRoutingEngine.0	Routing Engine compartment	1

Table 13 describes objects in the jnxContainersTable of an M20 router.

Table 13: jnxContainersEntry Objects in the jnxContainersTable of an M20 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM20.0	Chassis frame compartment	1
2	2	1	1	jnxM20SlotPower.0	Power supply compartment	2
4	3	1	1	jnxSlotFan.0	Fan compartment	4
6	2	1	1	jnxM20SlotSSB.0	SSB slot	2
7	1	1	1	jnxM20SlotFPC.0	FPC slot	4
8	1	2	7	jnxM20MediaCardSpace.PIC.0	PIC space	4
9	2	1	1	jnxM20RE.0	Routing Engine compartment	2
10	1	1	1	JNXM20FrontPanel.0	Front display slot	1

Table 14 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M160 router.

Table 14: jnxContainersEntry Objects in the jnxContainersTable of an M160 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM160.0	Chassis frame compartment	1
2	2	1	1	Jnx160SlotPower.0	Power supply compartment	2
4	3	1	1	jnxM160SlotFan.0	Fan compartment	4
6	2	1	1	jnxM160SlotSFM.0	SFM slot	4
7	1	1	1	jnxM160SlotFPC.0	FPC slot	8
8	1	2	7	jnxM160MediaCardSlotPIC.0	PIC space	4
9	2	1	1	jnxM160SlotHM.0	Host slot	2
10	1	1	1	jnxM160SlotFPM.0	FPM slot	1
11	2	1	1	jnxM160SlotPCG.0	PCG slot	2
12	2	1	1	jnxM160SlotMCS.0	MCS slot	2
13	1	1	1	jnxM160SlotCIP.0	CIP slot	1

Table 15 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M10 router.

Table 15: jnxContainersEntry Objects in the jnxContainersTable of an M10 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM10.0	Chassis frame compartment	1
2	2	1	1	jnxM10SlotPower.0	Power supply compartment	2
4	2	1	1	jnxM10SlotFan.0	Fan compartment	1
6	2	1	1	jnxM10SlotFEB.0	FEB slot	1
7	1	1	1	jnxM10SlotFPC.0	FPC slot	2
8	1	2	7	jnxM10MediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM10SlotRE.0	Routing Engine compartment	1

Table 16 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M5 router.

Table 16: jnxContainersEntry Objects in the jnxContainersTable of an M5 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM5.0	Chassis frame compartment	1
2	2	1	1	jnxM5SlotPower.0	Power supply compartment	2
4	3	1	1	jnxM5SlotFan.0	Fan compartment	4
6	2	1	1	jnxM5SlotFEB.0	FEB slot	1
7	1	1	1	jnxM5SlotFPC.0	FPC slot	1
8	1	2	7	jnxM5MediaCardSlotPIC.0	PIC space	4
9	2	1	1	jnxM5SlotRE.0	Routing Engine compartment	1

Table 17 describes objects contained in a jnxContainersEntry in the jnxContainersTable of a T640 routing node.

Table 17: jnxContainersEntry Objects in the jnxContainersTable of a T640 Routing Node

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisT640.0	Chassis frame	1
2	2	1	1	jnxT640SlotPower.0	PEM slot	2
4	3	1	1	jnxT640SlotFan.0	Fan slot	3
7	1	1	1	jnxT640SlotFPC.0	FPC slot	8
8	1	2	7	jnxT640MediaCardSpacePIC.0	PIC slot	4
9	2	1	1	jnxT640SlotHM.0	Host slot	2
10	1	1	1	jnxT640SlotFPB.0	FPM slot	1
11	2	1	1	jnxT640SlotSCG.0	SCG slot	2
12	2	1	1	jnxT640SlotCB.0	CG slot	2
13	1	1	1	jnxT640SlotCIP.0	CIP slot	1
14	2	1	1	jnxT640SlotSPMB.0	SPMB slot	2
15	2	1	1	jnxT640SlotSIB.0	SIB slot	5

Table 18 describes objects contained in a jnxContainersEntry in the jnxContainersTable of a T320 router.

Table 18: jnxContainersEntry Objects in the jnxContainersTable of a T320 Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisT320.0	Chassis frame	1
2	2	1	1	jnxT320SlotPower.0	PEM slot	2
4	3	1	1	jnx320SlotFan.0	Fan slot	3
7	1	1	1	jnxT320SlotFPC.0	FPC slot	8
8	1	2	7	jnxT320MediaCardSpacePIC.0	PIC slot	2

Index	View	Level	Within	Type	Description	Count
9	2	1	1	jnxT320SlotHM.0	Host slot	2
10	1	1	1	jnxT320SlotFPB.0	FPM slot	1
11	2	1	1	jnxT320SlotSCG.0	SCG slot	2
12	2	1	1	jnxT320SlotCB.0	CB slot	2
13	1	1	1	jnxT320SlotCIP.0	CIP slot	1
14	2	1	1	jnxT320SlotSPMB.0	SPMB slot	2
15	2	1	1	jnxT320SlotSIB.0	SIB slot	3

Table 19 describes objects contained in a jnxContainersEntry in the jnxContainersTable of an M40e router.

Table 19: jnxContainersEntry Objects in the jnxContainersTable of an M40e Router

Index	View	Level	Within	Type	Description	Count
1	1	0	0	jnxChassisM40e.0	Chassis frame compartment	1
2	2	1	1	jnxM40eSlotPower.0	Power supply compartment	2
4	3	1	1	jnxM40eSlotFan.0	Fan compartment	4
6	2	1	1	jnxM40eSlotSFM.0	SFM slot	2
7	1	1	1	jnxM40eSlotFPC.0	FPC slot	8
8	1	2	7	jnxM40eMediaCardSpacePIC.0	PIC space	4
9	2	1	1	jnxM40eSlotHM.0	Host slot	2
10	1	1	1	jnxM40eSlotFPM.0	FPM slot	1
11	2	1	1	jnxM40eSlotPCG.0	PCG slot	2
12	2	1	1	jnxM40eSlotMCS.0	MCS slot	2
13	1	1	1	jnxM40eSlotCIP.0	CIP slot	1

jnxContentsLastChange

The object identifier for jnxContentsLastChange object is {jnxBoxAnatomy 7}. This object indicates the time at which the box contents last changed, represented by the sysUpTime value.

jnxContentsTable

The object identifier for jnxContentsTable object is {jnxBoxAnatomy 8}. This object specifies the contents of the chassis.

The jnxContentsTable lists the contents of an entry, which are defined as follows:

- jnxContentsContainerIndex—Associates the jnxContainersIndex with the jnxContainersTable, whose object identifier is {jnxContentsEntry 1}.
- jnxContentsL1Index—The level-one index of the container housing the component, whose object identifier is {jnxContentsEntry 2}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.

MIBs start with a value of 1 while the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- jnxContentsL2Index—The level-two index of the container housing the component, whose object identifier is {jnxContentsEntry 3}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxContentsL3Index—The level-three index of the container housing the component, whose object identifier is {jnxContentsEntry 4}. It indicates the position of the component within different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxContentsType—The component at a specific container index or L1, L2, or L3 index, whose object identifier is {jnxContentsEntry 5}.
- jnxContentsDescr—The type of component described in plain English, whose object identifier is {jnxContentsEntry 6}.
- jnxContentsSerialNo—The serial number of the component, whose object identifier is {jnxContentsEntry 7}.
- jnxContentsRevision—The revision level of the component, whose object identifier is {jnxContentsEntry 8}.
- jnxContentsInstalled—The time at which the component was last installed and operational, represented by the sysUpTime value, whose object identifier is {jnxContentsEntry 9}.
- jnxContentsPartNo—The part number of the component (blank if unknown or unavailable), whose object identifier is {jnxContentsEntry 10}.

Table 20 through Table 22 provide examples of jnxContentEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxContentsEntry objects:

- Container index— jnxContentsContainerIndex
- L1—jnxContentsL1Index
- L2—jnxContentsL2Index
- L3—jnxContentsL3Index
- Type—jnxContentsType
- Description—jnxContentsDescr
- Serial Number—jnxContentsSerialNo
- Revision—jnxContentsRevision
- Installed—jnxContentsInstalled
- Part Number—jnxContentsPartNo

Table 20 provides an example of jnxContentEntry objects in the jnxContentTable of an M20 router.

Table 20: jnxContentsEntry Objects in the jnxContentsTable of an M20 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	1	0	jnxBackplaneM20.0	Midplane	AL3280	REV07	0:0:00:00.00	710-00157
2	1	0	0	jnxM20PowerDC.0	DC power supply A	001652	REV 05	0:0:00:00.00	740-00146
2	2	0	0	jnxM20PowerDC.0	DC power supply B	001652	REV 05	0:0:00:00.00	740-00146
4	1	0	0	jnxM20Fan.0	Front top fan			0:0:00:00.00	
4	2	0	0	jnxM20Fan	Middle fan			0:0:00:00.00	
4	3	0	0	jnxM20Fan	Bottom fan			0:0:00:00.00	
4	4	4	0	jnxM20Fan	Rear fan			0:0:00:00.00	
6	1	0	0	jnxM20SSB.0	SSB 0 Internet Processor II	AG0809	REV 01	0:0:00:35.17	710-001951
7	1	0	0	jnxM20FPC.0	FPC @ 0/*/*	AN1335	REV 01	0:0:01:01.80	710-001292
7	2	0	0	jnxM20FPC.0	FPC @ 1/*/*	AN1124	REV 01	0:0:01:07:96	710-001292
7	3	0	0	jnxM20FPC.0	FPC @ 2/*/*	AN1726	REV 01	0:0:01:14:12	710-001292
7	4	0	0	jnxM20FPC.0	FPC @ 3/*/*	AN1691	REV 01	0:0:01:20.28	710-001292
8	1	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/0/*	HD4313	REV 04	0:0:00:00.00	750-002992
8	1	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/1/*	AJ5844	REV 04	0:0:00:00.00	750-002992
8	1	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/2/*	HD4518	REV 04	0:0:00:00.00	750-002992
8	1	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 0/3/*	HD4515	REV 04	0:0:00:00.00	750-002992
8	2	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/0/*	HD4296	REV 04	0:0:00:00.00	750-002992
8	2	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/1/*	HD4323	REV 04	0:0:00:00.00	750-002992
8	2	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/2/*	HD4129	REV 04	0:0:00:00.00	750-002992

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
8	2	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 1/3/*	HD4341	REV 04	0:0:00:00.00	750-002992
8	3	1	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX@ 2/0/*	AH4147	REV 07	0:0:00:00.00	750-002303
8	3	2	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/1/*	AH4238	REV 07	0:0:00:00.00	750-002303
8	3	3	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/2/*	AH4116	REV 07	0:0:00:00.00	750-002303
8	3	4	0	jnxM20QuadEther.0	PIC: 4x, F/E, 100BASE-TX @ 2/3/*	AH4208	REV 07	0:0:00:00.00	750-002303
8	4	1	0	jnxM20GigEther.0	PIC: 1x G/E, 1000BASE-SX @ 3/0/*	AS3697	REV 07	0:0:00:00.00	750-001072
8	4	2	0	jnxM20ChOc12toDS 3.0	PIC: 1x COC12SMIR @ 3/1/*	AE1110	REV 08	0:0:00:00.00	750-001190
8	4	4	0	jnxM20ChStml1.0	PIC: 1x CSTM1SMIR @ 3/3/*	AD9599	REV 04	0:0:00:00.00	750-003250
9	1	0	0	jnxM20RE.0	Routing Engine			3:16:16:53.21	
10	1	0	0	jnxM20FrontPanel.0	Front panel display			0:0:00:00.00	

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M20 router is listed below.

```
user@host> show chassis hardware
Item            Version Part number Serial Number   Description
Chassis53711    M20
Backplane       REV 07  710-001517      AL3280
Power Supply A  REV 05  740-001466      001652      DC
Power Supply B  REV 05  740-001466      001632      DC
Display         REV 04  710-001519      AP9225
Host 0 c900000619e6ba01 teknor
SSB slot 0      REV 01  710-001951      AG0809      Internet Processor
II
FPC 0      REV 01  710-001292  AN1335
 PIC 0     REV 04  750-002992  HD4313      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  AJ5844      4x F/E, 100 BASE-TX
 PIC 2     REV 04  750-002992  HD4518      4x F/E, 100 BASE-TX
 PIC 3     REV 04  750-002992  HD4515      4x F/E, 100 BASE-TX
FPC 1      REV 01  710-001292  AN1124
 PIC 0     REV 04  750-002992  HD4296      4x F/E, 100 BASE-TX
 PIC 1     REV 04  750-002992  HD4323      4x F/E, 100 BASE-TX
 PIC 2     REV 04  750-002992  HD4129      4x F/E, 100 BASE-TX
 PIC 3     REV 04  750-002992  HD4341      4x F/E, 100 BASE-TX
FPC 2      REV 01  710-001292  AN1726
 PIC 0     REV 07  750-002303  AH4147      4x F/E, 100 BASE-TX
 PIC 1     REV 07  750-002303  AH4238      4x F/E, 100 BASE-TX
 PIC 2     REV 07  750-002303  AH4116      4x F/E, 100 BASE-TX
 PIC 3     REV 07  750-002303  AH4208      4x F/E, 100 BASE-TX
FPC 3      REV 01  710-001292  AN1691
 PIC 0     REV 08  750-001072  AS3697      1x G/E, 1000
BASE-SX
 PIC 1     REV 03  750-001190  AE1110      1x COC12, SMIR
 PIC 3     REV 04  750-003250  AD9599      1x CSTM1, SMIR
```

Table 21 provides an example of jnxContentEntry objects in the jnxContentTable of a T640 routing node.

Table 21: jnxContentsEntry Objects in the jnxContentsTable of a T640 Routing Node

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxMidplaneT640.0	Midplane	AX5633	REV 04	0:0:00:00.00	710-002726
2	2	0	0	jnxT640Power.0	PEM 1	MD21815	RevX02	0:0:00:00.00	740-002595
4	1	1	0	jnxT640Fan.0	Top Left Front fan			0:0:00:00.00	
4	1	2	0	jnxT640Fan.0	Top Left Middle fan			0:0:00:00.00	
4	1	3	0	jnxT640Fan.0	Top Left Rear fan			0:0:00:00.00	
4	1	4	0	jnxT640Fan.0	Top Right Front fan			0:0:00:00.00	
4	1	5	0	jnxT640Fan.0	Top Right Middle fan			0:0:00:00.00	
4	1	6	0	jnxT640Fan.0	Top Right Rear fan			0:0:00:00.00	
4	2	1	0	jnxT640Fan.0	Bottom Left Front fan			0:0:00:00.00	
4	2	2	0	jnxT640Fan.0	Bottom Left Middle fan			0:0:00:00.00	
4	2	3	0	jnxT640Fan.0	Bottom Left Rear fan			0:0:00:00.00	
4	2	4	0	jnxT640Fan.0	Bottom Right Front fan			0:0:00:00.00	
4	2	5	0	jnxT640Fan.0	Bottom Right Middle fan			0:0:00:00.00	
4	2	6	0	jnxT640Fan.0	Bottom Right Rear fan			0:0:00:00.00	
4	3	1	0	jnxT640Fan.0	Fourth Blower from top			0:0:00:00.00	
4	3	2	0	jnxT640Fan.0	Bottom Blower			0:0:00:00.00	
4	3	3	0	jnxT640Fan.0	Middle Blower			0:0:00:00.00	
4	3	4	0	jnxT640Fan.0	Top Blower			0:0:00:00.00	
4	3	5	0	jnxT640Fan.0	Second Blower from top			0:0:00:00.00	
7	2	0	0	jnxT640FPC.0	FPC @ 1/*/*	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	1	0	jnxT640FPC.0	FPC @ 1/0/* top temp sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	2	2	0	jnxT640FPC.0	FPC @ 1/1/* bottom temp sensor	HE3009	REV 01	0:18:56:48.81	710-002385
7	6	0	0	jnxT640FPC.0	FPC @ 5/*/*	HD5001	REV 03	0:18:57:02.71	710-001721
7	6	1	0	jnxT640FPC.0	FPC @ 5/0/* top temp sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	6	2	0	jnxT640FPC.0	FPC @ 5/1/* bottom temp sensor	HD5001	REV 03	0:18:57:02.71	710-001721
7	8	0	0	jnxT640FPC.0	FPC @ 7/*/*	HE3179	REV 01	0:18:56:52.85	710-002385
7	8	1	0	jnxT640FPC.0	FPC @ 7/0/* top temp sensor	HE3179	REV 01	0:18:56:52.85	710-002385
7	8	2	0	jnxT640FPC.0	FPC @ 7/1/* bottom temp sensor	HE3179	REV 01	0:18:56:52.85	710-002385
8	2	1	0	jnxT640PIC3.0	PIC: 1x G/E, 1000 BASE-SX @ 1/0/*	AP5542	REV 08	0:18:56:50.91	750-001072
8	2	2	0	jnxT640PIC3.0	PIC: 1x OC-12 ATM, SMIR @ 1/1/*	AK6894	REV 02	0:18:56:55.24	750-002983
8	2	3	0	jnxT640PIC3.0	PIC: 1x G/E, 1000 BASE-SX @ 1/2/*	HD4968	REV 04	0:18:56:55.64	750-001894
8	6	1	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/0/*	HC0273	REV 01	0:18:57:04.47	750-004535
8	6	2	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/1/*	HC0271	REV 01	0:18:57:04.55	750-004535
8	6	3	0	jnxT640PIC3.0	PIC: 1x OC-192 SM SR1 @ 5/2/*	HC0254	REV 01	0:18:57:04.64	750-004535
8	8	1	0	jnxT640PIC3.0	PIC: 2x G/E, 1000 BASE-SX @ 7/0/*	AD3632	REV 01	0:18:56:55.16	710-002381
8	8	2	0	jnxT640PIC3.0	PIC: 4x OC-12 SONET, SMIR @ 7/1/*	AD3831	REV 05	0:18:56:55.18	750-001901
8	8	3	0	jnxT640PIC3.0	PIC: 1x OC-48 SONET, SMIR @ 7/2/*	AA9603	REV 01	0:18:56:55.21	750-001900
8	8	4	0	jnxT640PIC3.0	PIC: 1x OC-48 SONET, SMSR @ 7/3/*	AD5724	REV 05	0:18:56:55.24	750-001900

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
9	1	0	0	jnxT640HM.0	Host 0			0:19:19:30.95	
9	2	0	0	jnxT640HM.0	Host 1	2108657 00292	REV 01	2:19:45:51.00	740-005022
10	1	0	0	jnxT640FPB.0	FPM	HE3245	REV 02	0:0:00:00.00	710-002901
11	1	0	0	jnxT640SCG.0	SCG 0	HF6023	REV 04	0:0:00:00.00	710-003423
11	2	0	0	jnxT640SCG.0	SCG 1	HF6061	REV 04	0:0:00:00.00	710-003423
12	2	0	0	jnxT640CB.0	CB 0	HE3614	REV 06	0:0:00:00.00	710-002728
12	2	0	0	jnxT640CB.0	CB 1	HE3627	REV 06	0:0:00:00.00	710-002728
13	1	0	0	jnxT640CIP.0	CIP	HA4729	REV 05	0:0:00:00.00	710-002895
14	1	0	0	jnxT640SPMB.0	SPMB 0	HF6876	REV 02	0:18:56:06.72	710-003229
14	2	0	0	jnxT640SPMB.0	SPMB 1	HG6237	REV 02	0:18:56:08.01	710-003229
15	1	0	0	jnxT640SIB.0	SIB 0	HJ9669	REV 02	0:0:00:00.00	710-005157
15	2	0	0	jnxT640SIB.0	SIB 1	HJ9668	REV 02	0:0:00:00.00	710-005157
15	3	0	0	jnxT640SIB.0	SIB 2	HH3039	REV 02	0:0:00:00.00	710-005157
15	4	0	0	jnxT640SIB.0	SIB 3	HH3041	REV 02	0:0:00:00.00	710-005157
15	5	0	0	jnxT640SIB.0	SIB 4	HJ9657	REV 02	0:0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item       Version Part number Serial number   Description
Chassis         T640
Midplane     REV 04  710-002726  AX5633
FPM GBUS     REV 02  710-002901  HE3245
FPM Display   REV 02  710-002897  HA4873
CIP       REV 05  710-002895  HA4729
PEM 1      RevX02  740-002595  MD21815      Power Entry Module
SCG 0      REV 04  710-003423  HF6023
SCG 1      REV 04  710-003423  HF6061
Host 0          unknown
Host 1   REV 01  740-005022  210865700292   RE-3.0
CB 0       REV 06  710-002728  HE3614
CB 1       REV 06  710-002728  HE3627
FPC 1      REV 01  710-002385  HE3009      FPC Type 1
 CPU       REV 06  710-001726  HC0010
 PIC 0     REV 08  750-001072  AP5542      1x G/E, 1000 BASE-SX
 PIC 1     REV 02  750-002983  AK6894      1x OC-12 ATM, SMIR
 PIC 2     REV 04  750-001894  HD4968      1x G/E, 1000 BASE-SX
 MMB 1     REV 03  710-001723  HE7264      MMB-144mbit
 ICBM      REV 01  710-003384  HE3042
 PPB 0     REV 01  710-003758  HE7173      PPB Type 2
 PPB 1     REV 01  710-003758  HE7170      PPB Type 2
FPC 5      REV 03  710-001721  HD5001      FPC Type 3
 CPU      REV 06  710-001726   HA5080
 PIC 0     REV 01  750-004535  HC0273      1x OC-192 SM SR1
 PIC 1     REV 01  750-004535  HC0271      1x OC-192 SM SR1
 PIC 2     REV 01  750-004535  HC0254      1x OC-192 SM SR1
 MMB 0     REV 03  710-001723  HE7263      MMB-144mbit
 MMB 1     REV 03  710-001723  HE7266      MMB-144mbit
 ICBM      REV 01  710-003384  HE3044
 PPB 0     REV 02  710-002845  HD6027      PPB Type 3
 PPB 1     REV 02  710-002845  HD6039      PPB Type 3
FPC 7      REV 01  710-002385  HE3179      FPC Type 2
 CPU      REV 06  710-001726   HE7915
 PIC 0     REV 01  710-002381  AD3632      2x G/E, 1000 BASE-SX
 PIC 1     REV 05  750-001901  AD3831      4x OC-12 SONET, SMIR
 PIC 2     REV 01  750-001900  AA9603      1x OC-48 SONET, SMIR
 PIC 3     REV 05  750-001900  AD5724      1x OC-48 SONET, SMSR
 MMB 1     REV 02  710-004047  HE3424      MMB-288mbit
 ICBM      REV 04  710-003384  HA4480
 PPB 0     REV 02  710-003758  HE3169      PPB Type 2
 PPB 1     REV 02  710-003758  HA4535      PPB Type 2
SPMB 0      REV 02  710-003229 HF6876
SPMB 1      REV 02  710-003229 HG6237
SIB 0      REV 02  710-005157  HJ9669      SIB-I8-F16
SIB 1      REV 02  710-005157  HJ9668      SIB-I8-F16
SIB 2      REV 02  710-005157  HH3039      SIB-I8-F16
SIB 3      REV 02  710-005157  HH3041      SIB-I8-F16
SIB 4      REV 02  710-005157  HJ9657      SIB-I8-F16
```

Table 22 provides an example of jnxContentEntry objects in the jnxContentTable of a T320 router.

Table 22: jnxContentsEntry Objects in the jnxContentsTable of a T320 Router

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
1	1	0	0	jnxMidplaneT320.0	Midplane	AY4527	Rev 01	(0) 0:00:00.00	710-004339
2	1	0	0	jnxT320Power.0	PEM 0	ML14099	Rev 01	(0) 0:00:00.00	
4	1	1	0	jnxT320Fan.0	Top Left Front fan			(0) 0:00:00.00	
4	1	2	0	jnxT320Fan.0	Top Left Middle fan			(0) 0:00:00.00	
4	1	3	0	jnxT320Fan.0	Top Left Rear fan			(0) 0:00:00.00	
4	1	4	0	jnxT320Fan.0	Top Right Front fan			(0) 0:00:00.00	
4	1	5	0	jnxT320Fan.0	Top Right Middle fan			(0) 0:00:00.00	
4	1	6	0	jnxT320Fan.0	Top Right Rear fan			(0) 0:00:00.00	
4	2	1	0	jnxT320Fan.0	Bottom Left Front fan			(0) 0:00:00.00	
4	2	2	0	jnxT320Fan.0	Bottom Left Middle fan			(0) 0:00:00.00	
4	2	3	0	jnxT320Fan.0	Bottom Left Rear fan			(0) 0:00:00.00	
4	2	4	0	jnxT320Fan.0	Bottom Right Front fan			(0) 0:00:00.00	
4	2	5	0	jnxT320Fan.0	Bottom Right Middle fan			(0) 0:00:00.00	
4	2	6	0	jnxT320Fan.0	Bottom Right Rear fan			(0) 0:00:00.00	
4	3	1	0	jnxT320Fan.0	Rear Tray Top fan			(0) 0:00:00.00	
4	3	2	0	jnxT320Fan.0	Rear Tray Second fan			(0) 0:00:00.00	
4	3	3	0	jnxT320Fan.0	Rear Tray Middle fan			(0) 0:00:00.00	
4	3	4	0	jnxT320Fan.0	Rear Tray Fourth fan			(0) 0:00:00.00	
4	3	5	0	jnxT320Fan.0	Rear Tray Bottom fan			(0) 0:00:00.00	
7	4	0	0	jnxT320FPC.0	FPC @ 3/*/*	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	1	0	jnxT320FPC.0	FPC @ 3/0/* top temp sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
7	4	2	0	jnxT320FPC.0	FPC @ 3/1/* bottom temp sensor	AY4706	REV 01	(26190949) 3 days, 0:45:09.49	710-004333
8	1	1	0	jnxT320PIC3	PIC: 1x OC-192 SM SR2 @ 0/0/*	HJ9283	REV 06	(6378) 0:01:03.78	750-004535
8	1	2	0	jnxT320PIC3	PIC: 1x OC-192 SM SR2 @ 0/1/*	HJ9298	REV 06	(6434) 0:01:04.34	750-004535
9	1	0	0	jnxT320HM.0	Host 0	210865700286	REV 01	(32762924) 3 days, 19:00:29.24	740-005022
9	2	0	0	jnxT320HM.0	Host 1	210929000186	REV 01	(110269900) 12 days, 18:18:19.00	740-005022
10	1	0	0	jnxT320FPB.0	FPM	AY4514	REV 02	(0) 0:00:00.00	710-004461
11	1	0	0	jnxT320SCG.0	SCG 0	AY4520	REV 06	(0) 0:00:00.00	710-004455
11	2	0	0	jnxT320SCG.0	SCG 1	AY4526	REV 06	(0) 0:00:00.00	710-004455
12	1	0	0	jnxT320CB.0	CB 0	AY4765	REV 11	(0) 0:00:00.00	710-002728

Container Index	L1 Index	L2 Index	L3 Index	Type	Description	Serial Number	Revision	Installed	Part Number
12	2	0	0	jnxT320CB.0	CB 1	HG6051	REV 06	(0) 0:00:00.00	710-002728
13	1	0	0	jnxT320CIP.0	CIP	HC0476	REV 05	(0) 0:00:00.00	710-002895
14	1	0	0	jnxT320SPMB.0	SPMB 0	HB1893	REV 02	(26186997) 3 days, 0:44:29.97	710-003229
14	2	0	0	jnxT320SPMB.0	SPMB 1	HD5520	REV 02	(26186913) 3 days, 0:44:29.13	710-003229
15	1	0	0	jnxT320SIB.0	SIB 0	BC1509	REV 02	(0) 0:00:00.00	710-005157
15	2	0	0	jnxT320SIB.0	SIB 1	BC1512	REV 02	(0) 0:00:00.00	710-005157
15	3	0	0	jnxT320SIB.0	SIB 2	BC1494	REV 02	(0) 0:00:00.00	710-005157

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T320 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis  T320
Midplane         REV 01   710-004339   AY4527
FPM GBUS         REV 02   710-004461   AY4514
FPM Display      REV 02   710-002897   HF6097
CIP              REV 05   710-002895   HC0476
PEM 0            Rev 01   740-004359   ML14099           Power Entry
Module
SCG 0            REV 06   710-004455   AY4520
SCG 1            REV 06   710-004455   AY4526
RE 0           REV 01   740-005022   210865700286      RE-3.0
RE 1           REV 01   740-005022   210929000186      RE-3.0
CB 0             REV 11   710-002728   AY4765
CB 1             REV 06   710-002728   HG6051
FPC 1            REV 01   710-004333   AY4507              FPC Type 3
  CPU            REV 06   710-001726   HA4719
  MMB 1          REV 03   710-004047   HD5738              MMB-288mbit
  PPB 0          REV 02   710-002845   HC0988              PPB Type 3
FPC 3            REV 01   710-004333   AY4706              FPC Type 3
  CPU            REV 06   710-001726   HE7916
  MMB 1          REV 03   710-004047   HG6326              MMB-288mbit
  PPB 0          REV 02   710-002845   HC0958              PPB Type 3
SPMB 0           REV 02   710-003229   HB1893
SPMB 1           REV 02   710-003229   HD5520
SIB 0            REV 02   710-005157   BC1509              SIB-I8-F16
SIB 1            REV 02   710-005157   BC1512              SIB-I8-F16
SIB 2            REV 02   710-005157   BC1494              SIB-I8-F16
```

jnxLEDLastChange

The object identifier for the jnxLEDLastChange object is {jnxBoxAnatomy 9}. This object indicates when the LED last changed state. Its value is 0 if the sysUpTime value is unknown, or if it already existed when the agent was active.

jnxLEDTable

The object identifier for the jnxLEDTable object is {jnxBoxAnatomy 10}. This object indicates the LED status of the router and lists the contents of an entry. Entries in the jnxLEDTable are represented by the jnxLEDEntry object, whose object identifier is {jnxLEDTable 1}.

The jnxLEDTable describes the components of the LED Box Indicators, whose elements are described as follows:

- jnxLEDAssociateTable—The associate table to which the entry is related, whose object identifier is {jnxLEDEntry 1}.
- jnxLEDAssociateIndex—The index of the subject in the associated table to which the entry is related, whose object identifier is {jnxLEDEntry 2}. The associate index is the index of the subject in the associated table, which returns you to the jnxContainersTable.
- jnxLEDL1Index—The level-one index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 3}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.

MIBs start with a value of 1, while the physical count on the router starts with a value of 0. To find the actual location of a component within a router, you must subtract 1 from the L1, L2, or L3 index.

- jnxLEDL2Index—The level-two index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 4}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxLEDL3Index—The level-three index of the associate table to which an entry is related, whose object identifier is {jnxLEDEntry 5}. It indicates the position of the component within the different levels of the containers. This value is 0 if the position is unavailable or not applicable.
- jnxLEDOriginator—The chassis component that originated the update, whose object identifier is {jnxLEDEntry 6}.

- jnxLEDDescr—The name or detailed description of the entry, whose object identifier is {jnxLEDEntry 7}.
- jnxLEDState—The state of the LED indicator, whose object identifier is {jnxLEDEntry 8}. The state can be any of the following:
  - Amber—Alarm, offline, not working
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Other—Unknown or unavailable
  - Red—Alert, component failed
  - Yellow—Alarm, warning
- jnxLEDStateOrdered—The state of the LED indicator, whose object identifier is {jnxLEDEntry 9}. jnxLEDStateOrdered provides the same information as jnxLEDState but lists the states in a different order. The state can be any of the following:
  - Blue—Online as the active primary
  - Green—Working normally online as a standby backup if there is an active primary
  - Amber—Alarm, offline, not working
  - Yellow—Alarm, warning
  - Red—Alert, component failed
  - Other—Unknown or unavailable

Table 23 through Table 25 provide examples of jnxLEDEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxLEDEntry objects:

- Associate table—jnxLEDAssociateTable
- L1—jnxLEDL1Index
- L2—jnxLEDL2Index
- L3—jnxLEDL3Index
- Originator—jnxLEDOriginator
- Description—jnxLEDDescr
- State—jnxLEDState

Table 23 provides an example of jnxLEDEntry objects in the jnxLEDTable of an M20 router.

Table 23: jnxLEDEntry Objects in the jnxLEDTable of an M20 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	jnxChassisM20.0	Chassis alarm LED	Other
jnxContentsTable	6	1	0	0	jnxM20SSB.0	SSB 1 LED	Blue
jnxContentsTable	6	2	0	0	jnxM20SSB.0	SSB 2 LED	Green
jnxContentsTable	7	1	0	0	jnxM20FPC.0	FPC 1 LED	Amber
jnxContentsTable	7	2	0	0	jnxM20FPC.0	FPC 2 LED	Blue
jnxContentsTable	7	3	0	0	jnxM20FPC.0	FPC 3 LED	Blue
jnxContentsTable	7	4	0	0	jnxM20FPC.0	FPC 4 LED	Amber
jnxContentsTable	9	1	0	0	jnxM20RE.0	Routing Engine 1 LED	Blue
jnxContentsTable	9	2	0	0	jnxM20RE.0	Routing Engine 2 LED	Other

Table 24 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T640 routing node.

Table 24: jnxLEDEntry Objects in the jnxLEDTable of a T640 Routing Node

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable	1	1	0	0	jnxChassisT640.0	Chassis alarm LED	Other
jnxContentsTable	7	1	0	0	jnxT640FPC.0	FPC slot 0 LED	Other
jnxContentsTable	7	2	0	0	jnxT640FPC.0	FPC slot 1 LED	Green
jnxContentsTable	7	3	0	0	jnxT640FPC.0	FPC slot 2 LED	Other
jnxContentsTable	7	4	0	0	jnxT640FPC.0	FPC slot 3 LED	Other
jnxContentsTable	7	5	0	0	jnxT640FPC.0	FPC slot 4 LED	Other
jnxContentsTable	7	6	0	0	jnxT640FPC.0	FPC slot 5 LED	Green
jnxContentsTable	7	7	0	0	jnxT640FPC.0	FPC slot 6 LED	Other
jnxContentsTable	7	8	0	0	jnxT640FPC.0	FPC slot 7 LED	Green
jnxContentsTable	9	1	0	0	jnxT640HM.0	Host 0 LED	Blue
jnxContentsTable	9	2	0	0	jnxT640HM.0	Host 1 LED	Green

Table 25 provides an example of jnxLEDEntry objects in the jnxLEDTable of a T320 router.

Table 25: jnxLEDEntry Objects in the jnxLEDTable of a T320 Router

Associate Table	Associate Index	L1 Index	L2 Index	L3 Index	Originator	Description	State
jnxContentsTable(3)	1	1	0	0	jnxChassisT320.0	Chassis alarm LED	Other
jnxContentsTable(3)	7	1	0	0	jnxT320FPC.0	FPC slot 0 LED	Other
jnxContentsTable(3)	7	2	0	0	jnxT320FPC.0	FPC slot 1 LED	Other
jnxContentsTable(3)	7	3	0	0	jnxT320FPC.0	FPC slot 2 LED	Other
jnxContentsTable(3)	7	4	0	0	jnxT320FPC.0	FPC slot 3 LED	Other
jnxContentsTable(3)	7	5	0	0	jnxT320FPC.0	FPC slot 4 LED	Other
jnxContentsTable(3)	7	6	0	0	jnxT320FPC.0	FPC slot 5 LED	Other
jnxContentsTable(3)	7	7	0	0	jnxT320FPC.0	FPC slot 6 LED	Other
jnxContentsTable(3)	7	8	0	0	jnxT320FPC.0	FPC slot 7 LED	Other
jnxContentsTable(3)	9	1	0	0	jnxT320HM.0	Host 0 LED	Blue
jnxContentsTable(3)	9	2	0	0	jnxT320HM.0	Host 1 LED	Green

jnxFilledLastChange

The object identifier for the jnxFilledLastChange object is {jnxBoxAnatomy 11}. This object indicates when the box filled status last changed. This variable is 0 if the sysUpTime value is unknown or it already existed when the agent was active.

jnxFilledTable

The object identifier for the jnxFilledTable object is {jnxBoxAnatomy 12}. This object indicates whether a specific container in the router is used (filled) or empty. This table is used for inventory and capacity planning.

Entries in the jnxFilledTable are represented by the jnxFilledEntry object, whose object identifier is {jnxFilledTable 1}.

The jnxFilledTable describes the status of specific containers whose component objects are described as follows:

- jnxFilledContainerIndex—The associated jnxContainersIndex in the jnxContainersTable, whose object identifier is {jnxFilledEntry 1}.
- jnxFilledL1Index—The level-one index of the container housing the entry, whose object identifier is {jnxFilledEntry 2}.
- jnxFilledL2Index—The level-two index of the container housing the entry, whose object identifier is {jnxFilledEntry 3}.

- jnxFilledL3Index—The level-three index of the container housing the entry, whose object identifier is {jnxFilledEntry 4}.
- jnxFilledDescr—The entry's name or detailed description of the entry, whose object identifier is {jnxFilledEntry 5}.
- jnxFilledState—The entry's state (filled or empty), whose object identifier is {jnxFilledEntry 6}.

Table 26 through Table 28 provide examples of jnxFilledEntry objects in the jnxFilledTable. The following column headings for each table are abbreviated to correspond to the parts of the jnxFilledEntry objects:

- Container index—jnxFilledContainerIndex
- L1—jnxFilledL1Index
- L2—jnxFilledL2Index
- L3—jnxFilledL3Index
- Description—jnxFilledDescr
- State—jnxFilledState

Table 26 provides an example of jnxFilledEntry objects in the jnxFilledTable of an M20 router.

Table 26: jnxFilledEntry Objects in the jnxFilledTable of an M20 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame compartment	Filled
1	1	1	0	Temperature sensor space 0	Filled
1	1	2	0	Temperature sensor space 1	Filled
2	1	0	0	Power supply compartment A	Filled
2	2	0	0	Power supply compartment B	Empty
3	1	0	0	Rear top impeller compartment	Filled
3	2	0	0	Front bottom impeller compartment	Filled
4	1	0	0	Rear left fan compartment	Filled
4	2	0	0	Right center fan compartment	Filled
4	3	0	0	Rear right fan compartment	Filled
5	1	0	0	Host controller compartment	Filled
6	1	0	0	SCB slot	Filled
7	1	0	0	FPC slot 0	Empty
7	2	0	0	FPC slot 1	Empty
7	3	0	0	FPC slot 2	Filled
7	4	0	0	FPC slot 3	Filled
7	5	0	0	FPC slot 4	Empty
7	6	0	0	FPC slot 5	Filled
7	7	0	0	FPC slot 6	Empty
7	8	0	0	FPC slot 7	Empty
8	1	1	0	PIC space @ 0/0/*	Empty
8	1	2	0	PIC space @ 0/1/*	Empty
8	1	3	0	PIC space @ 0/2/*	Empty
8	1	4	0	PIC space @ 0/3/*	Empty
8	2	1	0	PIC space @ 1/0/*	Empty
8	2	2	0	PIC space @ 1/1/*	Empty
8	2	3	0	PIC space @ 1/2/*	Empty
8	2	4	0	PIC space @ 1/3/*	Empty
8	3	1	0	PIC space @ 2/0/*	Filled
8	3	2	0	PIC space @ 2/1/*	Filled
8	3	3	0	PIC space @ 2/2/*	Filled
8	3	4	0	PIC space @ 2/3/*	Filled
8	4	1	0	PIC space @ 3/0/*	Filled
8	4	2	0	PIC space @ 3/1/*	Filled
8	4	3	0	PIC space @ 3/2/*	Filled
8	4	4	0	PIC space @ 3/3/*	Filled
8	5	1	0	PIC space @ 4/0/*	Empty
8	5	2	0	PIC space @ 4/1/*	Empty
8	5	3	0	PIC space @ 4/2/*	Empty

Container Index	L1	L2	L3	Description	State
8	5	4	0	PIC space @ 4/3/*	Empty
8	6	1	0	PIC space @ 5/0/*	Filled
8	6	2	0	PIC space @ 5/1/*	Filled
8	6	3	0	PIC space @ 5/2/*	Filled
8	6	4	0	PIC space @ 5/3/*	Filled
8	7	1	0	PIC space @ 6/0/*	Empty
8	7	2	0	PIC space @ 6/1/*	Empty
8	7	3	0	PIC space @ 6/2/*	Empty
8	7	4	0	PIC space @ 6/3/*	Empty
8	8	1	0	PIC space @ 7/0/*	Empty
8	8	2	0	PIC space @ 7/1/*	Empty
8	8	3	0	PIC space @ 7/2/*	Empty
8	8	4	0	PIC space @ 7/3/*	Empty
9	1	0	0	Routing Engine compartment	Filled

Table 27 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T640 routing node.

Table 27: jnxFilledEntry Objects in the jnxFilledTable of a T640 Routing Node

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Empty
2	2	0	0	PEM slot 1	Filled
4	1	1	0	Top Left Front fan slot	Filled
4	1	2	0	Top Left Middle fan slot	Filled
4	1	3	0	Top Left Rear fan slot	Filled
4	1	4	0	Top Right Front fan slot	Filled
4	1	5	0	Top Right Middle fan slot	Filled
4	1	6	0	Top Right Rear fan slot	Filled
4	2	1	0	Bottom Left Front fan slot	Filled
4	2	2	0	Bottom Left Middle fan slot	Filled
4	2	3	0	Bottom Left Rear fan slot	Filled
4	2	4	0	Bottom Right Front fan slot	Filled
4	2	5	0	Bottom Right Middle fan slot	Filled
4	2	6	0	Bottom Right Rear fan slot	Filled
4	3	1	0	Fourth Blower from top slot	Filled
4	3	2	0	Bottom Blower slot	Filled
4	3	3	0	Middle Blower slot	Filled
4	3	4	0	Top Blower slot	Filled
4	3	5	0	Second Blower from top slot	Filled
7	3	2	0	FPC slot 0	Empty

Container Index	L1	L2	L3	Description	State
7	3	3	0	FPC slot 0 top temp sensor	Empty
7	3	4	0	FPC slot 0 bottom temp sensor	Empty
7	3	5	0	FPC slot 1	Filled
7	3	6	0	FPC slot 1 top temp sensor	Filled
7	1	0	0	FPC slot 1 bottom temp sensor	Filled
7	1	1	0	FPC slot 2	Empty
7	1	2	0	FPC slot 2 top temp sensor	Empty
7	2	0	0	FPC slot 2 bottom temp sensor	Empty
7	2	1	0	FPC slot 3	Empty
7	2	2	0	FPC slot 3 top temp sensor	Empty
7	3	0	0	FPC slot 3 bottom temp sensor	Empty
7	3	1	0	FPC slot 4	Empty
7	3	2	0	FPC slot 4 top temp sensor	Empty
7	4	0	0	FPC slot 4 bottom temp sensor	Empty
7	4	1	0	FPC slot 5	Filled
7	4	2	0	FPC slot 5 top temp sensor	Filled
7	5	0	0	FPC slot 5 bottom temp sensor	Filled
7	5	1	0	FPC slot 6	Empty
7	5	2	0	FPC slot 6 top temp sensor	Empty
7	6	0	0	FPC slot 6 bottom temp sensor	Empty
7	6	1	0	FPC slot 7	Filled
7	6	2	0	FPC slot 7 top temp sensor	Filled
7	7	0	0	FPC slot 7 bottom temp sensor	Filled
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	1	3	0	PIC slot @ 0/2/*	Empty
8	1	4	0	PIC slot @ 0/3/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Filled
8	2	2	0	PIC slot @ 1/1/*	Filled
8	2	3	0	PIC slot @ 1/2/*	Filled
8	2	4	0	PIC slot @ 1/3/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	3	3	0	PIC slot @ 2/2/*	Empty
8	3	4	0	PIC slot @ 2/3/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Empty
8	4	2	0	PIC slot @ 3/1/*	Empty
8	4	3	0	PIC slot @ 3/2/*	Empty
8	4	4	0	PIC slot @ 3/3/*	Empty
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	5	3	0	PIC slot @ 4/2/*	Empty
8	5	4	0	PIC slot @ 4/3/*	Empty

Container Index	L1	L2	L3	Description	State
8	6	1	0	PIC slot @ 5/0/*	Filled
8	6	2	0	PIC slot @ 5/1/*	Filled
8	6	3	0	PIC slot @ 5/2/*	Filled
8	6	4	0	PIC slot @ 5/3/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	7	3	0	PIC slot @ 6/2/*	Empty
8	7	4	0	PIC slot @ 6/3/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Filled
8	8	2	0	PIC slot @ 7/1/*	Filled
8	8	3	0	PIC slot @ 7/2/*	Filled
8	8	4	0	PIC slot @ 7/3/*	Filled
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled
15	4	0	0	SIB slot 3	Filled
15	5	0	0	SIB slot 4	Filled

Table 28 provides an example of jnxFilledEntry objects in the jnxFilledTable of a T320 router.

Table 28: jnxFilledEntry Objects in the jnxFilledTable of a T320 Router

Container Index	L1	L2	L3	Description	State
1	1	0	0	Chassis frame	Filled
2	1	0	0	PEM slot 0	Filled
2	2	0	0	PEM slot 1	Empty
4	1	1	0	Top Left Front fan slot	Filled
4	1	2	0	Top Left Middle fan slot	Filled
4	1	3	0	Top Left Rear fan slot	Filled
4	1	4	0	Top Right Front fan slot	Filled
4	1	5	0	Top Right Middle fan slot	Filled
4	1	6	0	Top Right Rear fan slot	Filled
4	2	1	0	Bottom Left Front fan slot	Filled
4	2	2	0	Bottom Left Middle fan slot	Filled
4	2	3	0	Bottom Left Rear fan slot	Filled
4	2	4	0	Bottom Right Front fan slot	Filled
4	2	5	0	Bottom Right Middle fan slot	Filled
4	2	6	0	Bottom Right Rear fan slot	Filled
4	3	1	0	Rear Tray Top fan slot	Filled
4	3	2	0	Rear Tray Second fan slot	Filled
4	3	3	0	Rear Tray Middle fan slot	Filled
4	3	4	0	Rear Tray Fourth fan slot	Filled
4	3	5	0	Rear Tray Bottom fan slot	Filled
7	1	0	0	FPC slot 0	Empty
7	1	1	0	FPC slot 0 top temp sensor	Empty
7	1	2	0	FPC slot 0 bottom temp sensor	Empty
7	2	0	0	FPC slot 1	Empty
7	2	1	0	FPC slot 1 top temp sensor	Empty
7	2	2	0	FPC slot 1 bottom temp sensor	Empty
7	3	0	0	FPC slot 2	Empty
7	3	1	0	FPC slot 2 top temp sensor	Empty
7	3	2	0	FPC slot 2 bottom temp sensor	Empty
7	4	0	0	FPC slot 3	Filled
7	4	1	0	FPC slot 3 top temp sensor	Filled
7	4	2	0	FPC slot 3 bottom temp sensor	Filled
7	5	1	0	FPC slot 4	Empty
7	5	2	0	FPC slot 4 top temp sensor	Empty
7	5	0	0	FPC slot 4 bottom temp sensor	Empty
7	6	1	0	FPC slot 5	Empty
7	6	2	0	FPC slot 5 top temp sensor	Empty
7	6	0	0	FPC slot 5 bottom temp sensor	Empty
7	7	1	0	FPC slot 6	Empty

Container Index	L1	L2	L3	Description	State
7	7	2	0	FPC slot 6 top temp sensor	Empty
7	7	0	0	FPC slot 6 bottom temp sensor	Empty
7	8	1	0	FPC slot 7	Empty
7	8	2	0	FPC slot 7 top temp sensor	Empty
7	8	0	0	FPC slot 7 bottom temp sensor	Empty
8	1	1	0	PIC slot @ 0/0/*	Empty
8	1	2	0	PIC slot @ 0/1/*	Empty
8	2	1	0	PIC slot @ 1/0/*	Empty
8	2	2	0	PIC slot @ 1/1/*	Empty
8	3	1	0	PIC slot @ 2/0/*	Empty
8	3	2	0	PIC slot @ 2/1/*	Empty
8	4	1	0	PIC slot @ 3/0/*	Filled
8	4	2	0	PIC slot @ 3/1/*	Filled
8	5	1	0	PIC slot @ 4/0/*	Empty
8	5	2	0	PIC slot @ 4/1/*	Empty
8	6	1	0	PIC slot @ 5/0/*	Empty
8	6	2	0	PIC slot @ 5/1/*	Empty
8	7	1	0	PIC slot @ 6/0/*	Empty
8	7	2	0	PIC slot @ 6/1/*	Empty
8	8	1	0	PIC slot @ 7/0/*	Empty
8	8	2	0	PIC slot @ 7/1/*	Empty
9	1	0	0	Host 0 slot	Filled
9	2	0	0	Host 1 slot	Filled
10	1	0	0	FPM slot	Filled
11	1	0	0	SCG slot 0	Filled
11	2	0	0	SCG slot 1	Filled
12	1	0	0	CB slot 0	Filled
12	2	0	0	CB slot 1	Filled
13	1	0	0	CIP slot	Filled
14	1	0	0	SPMB slot 0	Filled
14	2	0	0	SPMB slot 1	Filled
15	1	0	0	SIB slot 0	Filled
15	2	0	0	SIB slot 1	Filled
15	3	0	0	SIB slot 2	Filled

jnxOperatingTable

The object identifier for jnxOperatingTable object is {jnxBoxAnatomy 13}. This object reports the operating status of various components such as CPU, buffers, and memory. Juniper Networks routers implement packet forwarding and routing functions with two separate components, the Packet Forwarding Engine and the Routing Engine, to ensure stability. The clean separation of these two functions permits superior forwarding performance and a highly reliable operating system. Therefore, it is not necessary to monitor CPU, memory, and buffer utilization, as is the case with traditional, monolithic code base routers. The Routing Engine has its own CPU, memory, and buffers—separate from those of the Packet Forwarding Engine. The ASIC-based Packet Forwarding Engine forwards packets on all interfaces at wire speed, eliminating the need to monitor packet buffers being exhausted. As a result, CPU utilization under 2 percent is normal.

Entries in the jnxOperatingTable are represented by the jnxOperatingEntry object, whose object identifier is {jnxOperatingTable 1}.

The jnxOperating Table describes the status of specific objects, which are described as follows:

- jnxOperatingContents—The associated jnxContentsIndex in the jnxContentsTable, whose object identifier is {jnxOperatingEntry 1}.
- jnxOperatingL1Index—The level-one index of the container housing the entry, whose object identifier is {jnxOperatingEntry 2}.
- jnxOperatingL2Index—The level-two index of the container housing the entry, whose object identifier is {jnxOperatingEntry 3}.
- jnxOperatingL3Index—The level-three index of the container housing the entry, whose object identifier is {jnxOperatingEntry 4}.
- jnxOperatingDescr—The name or detailed description of the entry, whose object identifier is {jnxOperatingEntry 5}.
- jnxOperatingState—The operating state of the entry, whose object identifier is {jnxOperatingEntry 6}. The state can be any of the following:
  - Unknown(1)—State of the component is unknown or unavailable
  - Running(2)—Up and running as an active primary
  - Ready(3)—Ready to run; not running yet
  - Reset(4)—Held in reset; not ready yet
  - RunningAtFullSpeed(5)—Valid for fans only
  - Down(6)—Power supply is down or off
  - Standby(7)—Running as a standby backup
- jnxOperatingTemp—The entry's temperature, in degrees Celsius (°C), whose object identifier is {jnxOperatingEntry 7}.

- jnxOperatingCPU—The CPU utilization percentage of the entry, whose object identifier is {jnxOperatingEntry 8}. It is valid for the control board, the FPC, and the Routing Engine. It is a five-second rolling weighted average calculated every second for each of the CPUs. The value is sent to the Routing Engine every 10 seconds. The value for the Routing Engine is an average of samples taken every 30 seconds over a five-minute period. jnxOperatingCPU.9.1.0.0. is for the Routing Engine CPU. The Routing Engine is the only object of interest; the rest are most likely zero because CPUs on those cards are only used for management purposes.

- jnxOperatingISR—The CPU utilization percentage of the entry in relation to the interrupt service routing (ISR), whose object identifier is {jnxOperatingEntry 9}.

- jnxOperatingDRAMSize—The DRAM size of the entry in bytes, whose object identifier is {jnxOperatingEntry 10}. It is valid for the FPC, Routing Engine, and control board.

- jnxOperatingBuffer—The buffer pool utilization of the entry (a percentage), whose object identifier is {jnxOperatingEntry 11}. It is valid for the FPC and control board as a percentage of utilization. Buffers are normally fixed-length memory preallocated for read/write, input/output, or reception/transmission. A measurement against these buffers gives some indication of how busy the system is. The larger the percentage utilization, the busier the system. In terms of absolute numbers, the bigger the buffer size, the better the system can handle bursty traffic patterns.

- jnxOperatingHeap—The heap utilization of the entry, whose object identifier is {jnxOperatingEntry 12}.

- jnxOperatingUpTime—The time interval, in 10-millisecond periods, that the entry has been up and running, whose object identifier is {jnxOperatingEntry 13}.

- jnxOperatingLastRestart—The value of sysUpTime when the entry was last restarted, whose object identifier is {jnxOperatingEntry 14}.

- jnxOperatingMemory—The entry's installed memory size in megabytes (MB), whose object identifier is {jnxOperatingEntry 15}.

- jnxOperatingStateOrdered—The operating state of the entry, whose object identifier is {jnxOperatingEntry 16}. The state can be any of the following

  - Running(1)—Up and running as an active primary
  - Standby(2)—Running as a standby backup
  - Ready(3)—Ready to run; not running yet
  - RunningAtFullSpeed(4)—Valid for fans only
  - Reset(5)—Held in reset; not ready yet
  - Down(6)—Power supply is down or off
  - Unknown(7)—State of the component is unknown or unavailable

Table 29 through Table 31 provide examples of jnxOperatingEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxOperatingEntry objects:

- Contents index—jnxOperatingContents
- L1—jnxOperatingL1Index
- L2—jnxOperatingL2Index
- L3—jnxOperatingL3Index
- Description—jnxOperatingDescr
- State—jnxOperatingState
- Temp—jnxOperatingTemp
- CPU—jnxOperatingCPU
- ISR—jnxOperatingISR
- DRAM—jnxOperatingDRAMSize
- Buffer—jnxOperatingBuffer
- Heap—jnxOperatingHeap
- UpTime—jnxOperatingUpTime
- Last Restart—jnxOperatingLastRestart
- Memory—jnxOperatingMemory

Table 29 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of an M20 router.

Table 29: jnxOperatingEntry Objects in the jnxOperatingTable of an M20 Router

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
1	1	0	0	Midplane	Running	26	0	0	0	0	0	0	0:0:00:00.0	0
2	1	0	0	Power supply A	Running	28	0	0	0	0	0	0	0:0:00:00.0	0
2	2	0	0	Power supply B	Running	29	0	0	0	0	0	0	0:0:00:00.0	0
4	1	0	0	Front top fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	2	0	0	Front middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	3	0	0	Front bottom fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
4	4	0	0	Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.0	0
6	1	0	0	SSB 0	Running	30	0	0	67108864	6	0	67038195	0:0:00:35.41	64
7	1	0	0	FPC @ 0/*/*	Running	31	0	0	8388608	3	0	67035034	0:0:01:06.91	8

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
7	2	0	0	FPC @ 1/*/*	Running	33	0	0	8388608	4	0	67034422	0:0:01:13.04	8
7	3	0	0	FPC @ 2/*/*	Running	31	0	0	8388608	3	0	67033809	0:0:01:19.18	8
9	1	0	0	Routing Engine 0	Running	29	4	0	802738176	0	0	67046146	0:0:00:00.00	765

To verify the size of the memory, use the show chassis fpc, show chassis routing-engine, and show chassis ssb commands. For more information on the output of these commands, see the *JUNOS Internet Software Operational Mode Command Reference.*

Table 30 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of a T640 routing node.

Table 30: jnxOperatingEntry Objects in the jnxOperatingTable of a T640 Routing Node

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
1	1	0	0	Midplane	Running	0								
2	2	0	0	PEM 1	Running	29								
4	1	1	0	Top Left Front fan	Running	0								
4	1	2	0	Top Left Middle fan	Running	0								
4	1	3	0	Top Left Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	4	0	Top Right Front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	5	0	Top Right Middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	1	6	0	Top Right Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	1	0	Bottom Left Front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	2	0	Bottom Left Middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	3	0	Bottom Left Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	4	0	Bottom Right Front fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	2	5	0	Bottom Right Middle fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom Right Rear fan	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	1	0	Bottom Blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	2	0	Bottom Blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	3	0	Middle Blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
4	3	4	0	Top Blower	Running	0	0	0	0	0	0	0	0:0:00:00.00	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
4	3	5	0	Second Blower from top	Running	0	0	0	0	0	0	0	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	Running	0	1	0	512	41	3	138367	0:18:56:48.81	512
7	2	1	0	FPC @ 1/0/* top temp sensor	Running	35	0	0	0	0	0	0	0:18:56:48.81	0
7	2	2	0	FPC @ 1/1/* bottom temp sensor	Running	32	0	0	0	0	0	0	0:18:56:48.81	0
7	6	0	0	FPC @ 5/*/*	Running	0	3	0	256	41	14	136976	0:18:57:02.71	256
7	6	1	0	FPC @ 5/0/* top temp sensor	Running	44	0	0	0	0	0	0	0:18:57:02.71	0
7	6	2	0	FPC @ 5/1/* bottom temp sensor	Running	33	0	0	0	0	0	0	0:18:57:02.71	0
7	8	0	0	FPC @ 7/*/*	Running	0	2	0	256	41	7	137963	0:18:56:52.85	256
7	8	1	0	FPC @ 7/0/* top temp sensor	Running	38	0	0	0	0	0	0	0:18:56:52.85	0
7	8	2	0	FPC @ 7/1/* bottom temp sensor	Running	33	0	0	0	0	0	0	0:18:56:52.85	0
9	1	0	0	Host 0	Running	35	0	0	2048	0	0	6963005	0:19:20:30.07	2048
9	2	0	0	Host 1	Standby	32	2	0	2048	0	0	24401100	2:19:46:51.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	0:0:00:00.00	0
11	1	0	0	SCG 0	Running	36	0	0	0	0	0	0	0:0:00:00.00	0
11	2	0	0	SCG 1	Standby	35	0	0	0	0	0	0	0:0:00:00.00	0
12	1	0	0	CB 0	Running	36	0	0	0	0	0	0	0:0:00:00.00	0
12	2	0	0	CB 1	Standby	39	0	0	0	0	0	0	0:0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	1	0	128	40	0	142576	0:18:56:06.72	128
14	2	0	0	SPMB 1	Standby	39	0	0	128	40	0	142447	0:18:56:08.01	128
15	1	0	0	SIB 0	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0
15	2	0	0	SIB 1	Unknown	39	0	0	0	0	0	0	0:0:00:00.00	0
15	3	0	0	SIB 2	Unknown	39	0	0	0	0	0	0	0:0:00:00.00	0
15	4	0	0	SIB 3	Unknown	40	0	0	0	0	0	0	0:0:00:00.00	0
15	5	0	0	SIB 4	Unknown	40	0	0s	0	0	0	0	0:0:00:00.00	0

Table 31 provides an example of jnxOperatingEntry objects in the jnxOperatingTable of a T320 router.

Table 31: jnxOperatingEntry Objects in the jnxOperatingTable of a T320 Router

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
1	1	0	0	Midplane	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
2	1	0	0	PEM 0	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	1	0	Top Left Front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	2	0	Top Left Middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	3	0	Top Left Rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	4	0	Top Right Front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	1	5	0	Top Right Middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Top Right Rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	1	0	Bottom Left Front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	2	0	Bottom Left Middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	3	0	Bottom Left Rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	4	0	Bottom Right Front fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	5	0	Bottom Right Middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	2	6	0	Bottom Right Rear fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	1	0	Rear Tray Top fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	2	0	Rear Tray Second fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	3	0	Rear Tray Middle fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	4	0	Rear Tray Fourth fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
4	3	5	0	Rear Tray Bottom fan	Running	0	0	0	0	0	0	0	(0) 0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	Running	0	1	0	256	41	7	6568428	(26190949) |3 days, 0:45:09.49	256
7	4	1	0	FPC @ 3/0/* top temp sensor	Running	41	0	0	0	0	0	0	26190949) 3 days, 0:45:09.49	0
7	4	2	0	FPC @ 3/1/* bottom temp sensor	Running	37	0	0	0	0	0	0	(26190949) 3 days, 0:45:09.49	0

Contents Index	L1	L2	L3	Description	State	Temp	CPU	ISR	DRAM	Buffer	Heap	UpTime	Last Restart	Memory
9	1	0	0	Host 0	Running	34	1	0	2048	0	0	32763001	(32763004) 3 days, 19:00:30.04	2048
9	2	0	0	Host 1	Standby	32	1	0	2048	0	0	110271900	(110271900) 12 days, 18:18:39.00	2048
10	1	0	0	FPM	Running	30	0	0	0	0	0	0	(0) 0:00:00.00	0
11	1	0	0	SCG 0	Running	33	0	0	0	0	0	0	(0) 0:00:00.00	0
11	2	0	0	SCG 1	Standby	31	0	0	0	0	0	0	(0) 0:00:00.00	0
12	1	0	0	CB 0	Running	37	0	0	0	0	0	0	(0) 0:00:00.00	0
12	2	0	0	CB 1	Standby	34	0	0	0	0	0	0	(0) 0:00:00.00	0
14	1	0	0	SPMB 0	Running	36	0	0	128	40	0	6572381	(26186997) 3 days, 0:44:29.97	128
14	2	0	0	SPMB 1	Standby	36	1	0	128	40	0	6572465	(26186913) 3 days, 0:44:29.13	128
15	1	0	0	SIB 0	Standby	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	2	0	0	SIB 1	Running	36	0	0	0	0	0	0	(0) 0:00:00.00	0
15	3	0	0	SIB 2	Running	38	0	0	0	0	0	0	(0) 0:00:00.00	0

jnxRedundancyTable

The object identifier for the jnxRedundancyTable is {jnxBoxAnatomy 14}. This object shows the internal configuration settings for the redundant subsystems or components in the chassis.

Entries within the jnxRedundancyTable are represented by the jnxRedundancyEntry object, whose object identifier is {jnxRedundancyEntry 1}. This jnxRedundancyEntry contains the following objects, which describe the internal configuration settings for the redundant subsystems or components in the chassis:

- jnxRedundancyContentsIndex—The index value of an entry in jnxRedundancyEntry, whose object identifier is {jnxContainersEntry 1}.
- jnxRedundancyL1Index—The level-one index associated with the redundant component, whose object identifier is {jnxContainersEntry 2}.
- jnxRedundancyL2Index—The level-two index associated with the redundant component, whose object identifier is {jnxContainersEntry 3}.
- jnxRedundancyL3Index—The level-three index associated with the redundant component, whose object identifier is {jnxContainersEntry 4}.
- jnxRedundancyDescr—The description of the redundant component, whose object identifier is {jnxContainersEntry 5}.
- jnxRedundancyConfig—The election priority of redundancy configuration, whose object identifier is {jnxContainersEntry 6}.
- jnxRedundancyState—The current running state of the redundant component, whose object identifier is {jnxContainersEntry 7}.

- jnxRedundancySwitchoverCount—The total number of switchovers, defined as a change in the jnxRedundancyState from master to backup or vice versa, as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is {jnxContainersEntry 8}.
- jnxRedundancySwitchoverTime—The value of sysUpTime when the jnxRedundancyState was last switched over from master to backup or vice versa, whose object identifier is {jnxContainersEntry 9}.
- jnxRedundancySwitchoverReason—The reason for the last switchover to the redundant component, whose object identifier is {jnxContainersEntry 10}.
- jnxKeepaliveHeartbeat—The period of sending keepalive messages between the master and the backup subsystem, which is a systemwide preset value in seconds used by internal mastership resolution, whose object identifier is {jnxContainersEntry 11}.
- jnxRedundancyKeepaliveTimeout—The timeout period in seconds used by the watchdog timer before it initiates a switchover to the backup subsystem, whose object identifier is {jnxContainersEntry 12}.
- jnxRedundancyKeepaliveElapsed—The elapsed time since the redundant component received the last keepalive message from the outer subsystems, whose object identifier is {jnxContainersEntry 13}.
- jnxRedundancyKeepaliveLoss—The total number of keepalive messages lost between the master and the backup subsystems as perceived by the redundant component since the Routing Engine is up and running, whose object identifier is {jnxContainersEntry 14}.

Table 32 through Table 34 provide examples of jnxRedundancyEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxOperatingTable objects:

- Contents index—jnxRedundancyContentsIndex
- L1—jnxRedundancyL1Index
- L2—jnxRedundancyL2Index
- L3—jnxRedundancyL3Index
- Description—jnxRedundancyDescr
- Config—jnxRedundancyConfig
- State—jnxRedundancyState
- Count—jnxRedundancySwitchoverCount
- Time—jnxRedundancySwitchoverTime
- Reason—jnxRedundancySwitchoverReason
- Heartbeat—jnxKeepaliveHeartbeat
- Timeout—jnxRedundancyKeepaliveTimeout

- Elapsed—jnxRedundancyKeepaliveElapsed
- Loss—jnxRedundancyKeepaliveLoss

Table 32 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of an M20 router.

Table 32: jnxRedundancyEntry Objects in the jnxRedundancyTable of an M20 Router

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heartbeat	Timeout	Elapsed	Loss
6	1	0	0	SSB 0 Internet Processor II	master	master	0	3383	never switched	0	0	0	0
6	2	0	0	SSB 1	disabled	disabled	0	0	never switched	0	0	0	0
9	1	0	0	Routing Engine 0	master	master	1	421	user switched	3	300	1	0
9	2	0	0	Routing Engine 1	backup	backup	0	0	others	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M20 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
 Slot 0:
  Current state         Master
  Election priority       Master (default)
  Temperature         26 degrees C / 78 degrees F
  DRAM            768 Mbytes
  CPU utilization:
   User            2 percent
   Background         0 percent
   Kernel           0 percent
   Interrupt         0 percent
   Idle           98 percent
  Model             teknor
  Serial ID           32000004f8ff1201
  Start time           2002-01-29 12:30:42 PST
  Uptime             21 hours, 17 minutes, 14 seconds
  Load averages:         1 minute  5 minute 15 minute
    0.03    0.02    0.00
Routing Engine status:
 Slot 1:
  Current state         Backup
  Election priority       Backup (default)
  DRAM           805306368 Mbytes
  CPU utilization:
   User            0 percent
   Background         0 percent
   Kernel           1 percent
   Interrupt         0 percent
   Idle           99 percent
  Model             teknor
  Serial ID           100000078c10df01
  Start time           2002-01-24 16:47:39 PST
  Uptime             5 days, 17 hours, 14 seconds
```

To verify SSB status, use the show chassis ssb command. Sample command output from an M20 router is listed below.

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
 State                       Master
 Temperature                 24 degrees C / 75 degrees F
 CPU utilization             2 percent
 Interrupt utilization       0 percent
 Heap utilization            16 percent
 Buffer utilization          43 percent
 Total CPU DRAM              64 Mbytes
 Internet Processor II       Version 1, Foundry IBM, Part number 9
 Start time:                 2002-01-29 12:32:24 PST
 Uptime:                     21 hours, 30 minutes, 53 seconds
Slot 1 information:
 State                       Backup
```

Table 33 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of a T640 routing node.

Table 33: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T640 Routing Node

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heartbeat	Timeout	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	3	0:18:55:49.42	User switched	20	300	1	0
9	2	0	0	Host 1	Backup	Backup	0	0:0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Unknown	Backup	1	0:0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	4	0	0	SIB 3	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0
15	5	0	0	SIB 4	Unknown	Master	1	0:0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state  Master
    Election priority             Master (default)
    Temperature 35 degrees C / 95 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       1 percent
      Background 0 percent
      Kernel     5 percent
      Interrupt  0 percent
      Idle      94 percent
    Model          unknown
    Start time     2002-03-31 14:26:49 PST
    Uptime         19 hours, 22 minutes, 13 seconds
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
```

```
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority             Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
    Start time     2002-03-29 14:00:18 PST
    Uptime         2 days, 19 hours, 48 minutes, 32 seconds
```

Table 34 provides an example of jnxRedundancyEntry objects in the jnxRedundancyTable of a T320 router.

Table 34: jnxRedundancyEntry Objects in the jnxRedundancyTable of a T320 Router

Content Index	L1	L2	L3	Description	Config	State	Count	Time	Reason	Heartbeat	Timeout	Elapsed	Loss
9	1	0	0	Host 0	Master	Master	6	(26185188) 3 days, 0:44:11.88	User switched	20	300	1	0
9	2	0	0	Host 1	Backup	Backup	0	(0) 0:00:00.00	Other	0	0	0	0
15	1	0	0	SIB 0	Backup	Backup	1	(0) 0:00:00.00	0	0	0	0	0
15	2	0	0	SIB 1	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0
15	3	0	0	SIB 2	Master	Master	1	(0) 0:00:00.00	0	0	0	0	0

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from a T320 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state  Master
    Election priority             Master (default)
    Temperature 34 degrees C / 93 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     1 percent
      Interrupt  0 percent
      Idle      98 percent
    Model          RE-3.0
    Start time     2002-04-05 14:43:16 PST
    Uptime         17 days, 23 hours, 3 minutes, 47
seconds
    Load averages: 1 minute   5 minute  15 minute
       0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state  Backup
    Election priority             Backup (default)
    Temperature 32 degrees C / 89 degrees F
    DRAM      2048 MB
    CPU utilization:
      User       0 percent
      Background 0 percent
      Kernel     0 percent
      Interrupt  0 percent
      Idle     100 percent
    Model          RE-3.0
    Start time     2002-03-27 15:25:07 PST
    Uptime         26 days, 22 hours, 21 minutes, 44 seconds
```

jnxFruTable

The object identifier for the jnxFruTable is {jnxBoxAnatomy 15}. This object shows the status of Field-Replaceable Units (FRUs) in the chassis.

Entries within the jnxFruTable are represented by the jnxFruEntry object, whose object identifier is {jnxFruEntry 1}. This jnxFruEntry object contains the following objects, which describe the FRUs in the chassis:

- jnxFruContentsIndex—The index value of an entry in jnxFruEntry, whose object identifier is {jnxFruEntry 1}.
- jnxFruL1Index—The level-one index associated with the FRU, whose object identifier is {jnxFruEntry 2}.
- jnxFruL2Index—The level-two index associated with the FRU, whose object identifier is {jnxFruEntry 3}.
- jnxFruL3Index—The level-three index associated with the FRU, whose object identifier is {jnxFruEntry 4}.

- jnxFruName—The name or detailed description of the FRU, whose object identifier is {jnxFruEntry 5}.
- jnxFruType—The FRU type, whose object identifier is {jnxFruEntry 6}. The FRU type can be any of the following:
  - other(1)
  - clockGenerator(2)
  - flexiblePicConcentrator(3)
  - switchingAndForwardingModule(4)
  - controlBoard(5)
  - routingEngine(6)
  - powerEntryModule(7
  - frontPanelModule(8)
  - switchInterfaceBoard(9)
  - processorMezzanineBoardForSIB(10)
  - portInterfaceCard(11)
  - craftInterfacePanel(12)
  - fan(13)
- jnxFruSlot—The slot number of the FRU, whose object identifier is {jnxFruEntry 7}. This is equivalent to jnxFruL1Index. The slot number is zero if unavailable or inapplicable.
- jnxFruState—The current state of the FRU, whose object identifier is {jnxFruEntry 8}. The FRU state can be any of the following:
  - unknown(1),
  - empty(2),
  - present(3),
  - ready(4),
  - announceOnline(5),
  - online(6),
  - announceOffline(7),
  - offline(8),
  - diagnostic(9),
  - standby(10)

- jnxFruTemp—The temperature in degrees Celsius of the FRU, whose object identifier is {jnxFruEntry 9}. The value is zero if unavailable or inapplicable.
- jnxFruOfflineReason—The reason the FRU is offline, whose object identifier is {jnxFruEntry 10}. The reason can be any of the following:
  - unknown(1)—Unknown or other
  - none(2)—None
  - error(3)—Error
  - noPower(4)—No power
  - configPowerOff(5)—Configured to power off
  - configHoldInReset(6)—Configured to hold in reset
  - cliCommand(7)—Brought offline by CLI command
  - buttonPress(8)—Brought offline by button press
  - cliRestart(9)—Restarted by CLI command
  - overtempShutdown(10)—Overtemperature shutdown
  - masterClockDown(11)—Master clock down
  - singleSfmModeChange(12)—Single SFM mode change
  - packetSchedulingModeChange(13)—Packet scheduling mode change
  - physicalRemoval(14)—Physical removal
  - unresponsiveRestart(15)—Restarting unresponsive board
  - sonetClockAbsent(16)—SONET out clock absent
- jnxFruLastPowerOff—The value of sysUpTime when this subject was last powered off, whose object identifier is {jnxFruEntry 11}. The value is zero if unavailable or inapplicable.
- jnxFruLastPowerOn—The value of sysUpTime when this subject was last powered on, whose object identifier is {jnxFruEntry 12}. The value is zero if unavailable or inapplicable.
- jnxFruPowerUpTime—The time interval in 10-millisecond periods that this subject has been up and running since the last power-on time, whose object identifier is {jnxFruEntry 13}. The value is zero if unavailable or inapplicable.

Table 35 through Table 40 provide examples of jnxFruEntry objects. The following column headings for each table are abbreviated to correspond to the parts of the jnxFruEntry objects:

- Contents Index—jnxFruContentsIndex
- L1—jnxFruL1Index
- L2—jnxFruL2Index
- L3—jnxFruL3Index
- Name—jnxFruName
- Type—jnxFruType
- Slot—jnxFruSlot
- State—jnxFruState
- Temp—jnxFruTemp
- Offline—jnxFruOffline
- PowerOff—jnxFruPowerOff
- PowerOn—jnxFruPowerOn
- Uptime—jnxFruPowerUpTime

Table 35 provides an example of jnxFruContent objects in the jnxFruTable for an M10 router.

Table 35: jnxFruContents Objects in the jnxFruTable of an M10 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power Supply A	powerEntryModule	1	online	0	none	0:0:00:00.00	0:0:11:08.73	26431910
2	2	0	0	Power Supply B	powerEntryModule	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	1	0	Left Fan 1	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Left Fan 2	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Left Fan 3	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Left Fan 4	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	FEB Internet Processor II	controlBoard	1	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x Monitor @ 0/1/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 1x OC-12 ATM, MM @ 0/2/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 4x T3 @ 0/3/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	1	0	PIC: 4x OC-3 SONET, SMIR @ 1/0/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 2x OC-3 ATM, MM @ 1/3/*	portInterfaceCard	2	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M10 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                58974             M10
Midplane         REV 03   710-001950   HB1590
Power Supply A   Rev 03   740-002498   LK33505           DC
Display          REV 04   710-001995   HE8442
Routing Engine   REV 01   740-003239   9001025728        RE-2.0
FEB              REV 12   710-001948   HA4221            Internet Processor II
FPC 0
  PIC 1          REV 01   750-004188   AR2912            1x Monitor
  PIC 2          REV 04   750-001551   AN7869            1x OC-12 ATM, MM
  PIC 3          REV 02   750-002485   AN2803            4x T3
FPC 1
  PIC 0          REV 03   750-002970   HF2293            4x OC-3 SONET, SMIR
  PIC 1          REV 03   750-002971   HA8094            4x OC-3 SONET, MM
  PIC 2          REV 03   750-002977   HD9352            2x OC-3 ATM, MM
  PIC 3          REV 03   750-002977   HD9393            2x OC-3 ATM, MM
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M10 router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            24      3          1        64        44         17
  1  Online            24      3          1        64        44         17
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M10 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                 26 degrees C / 78 degrees F
    DRAM                       768 MB
    Memory utilization           9 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     0 percent
      Interrupt                  0 percent
      Idle                     100 percent
    Model                        RE-2.0
    Serial ID                    b7000007c81ce801
    Start time                   2002-06-21 09:33:45 PDT
    Uptime                       3 days, 1 hour, 23 minutes, 27 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.07       0.03       0.01
```

To verify FEB status, use the show chassis feb command. Sample command output from an M10 router is listed below.

```
user@host> show chassis feb
FEB status:
  Temperature                24 degrees C / 75 degrees F
  CPU utilization             3 percent
  Interrupt utilization       1 percent
  Heap utilization           17 percent
  Buffer utilization         44 percent
  Total CPU DRAM             64 MB
  Internet Processor II         Version 1, Foundry IBM, Part number 9
  Start time:                   2002-06-21 09:45:46 PDT
  Uptime:                       3 days, 1 hour, 11 minutes, 33 seconds
```

Table 36 provides an example of jnxFruContent objects in the jnxFruTable for an M20 router.

Table 36: JnxFruContents Objects in the jnxFruTable of an M20 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power Supply A	powerEntryModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	Power Supply B	powerEntryModule	2	online	25	none	0:0:00:00.00	0:0:00:43.45	24993357
4	1	0	0	Rear Fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Front Upper Fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Front Middle Fan	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Front Bottom Fan	fan	4	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SSB 0	controlBoard	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	2	0	0	SSB 1 Internet Processor I	controlBoard	2	online	29	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	27	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	27	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	offline	28	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x Tunnel @ 1/0/*	portInterfaceCard	2	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x T3 @ 1/1/*	portInterfaceCard	2	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x OC-3 ATM, MM @ 1/2/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x G/E, 1000 BASE-SX @ 1/3/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	ready	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 2x OC-3 SONET, SMIR @ 3/2/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	online	25	none	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	2	online	24	none	0:0:00:00.00	0:0:00:00.00	0
10	1	0	0	Front Panel Display	frontPanelModule	1	online	0	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M20 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                20200             M20
Backplane        REV 07   710-001517   AB5911
Power Supply B   Rev 02   7            000240            AC
Display          REV 04   710-001519   AD1903
Routing Engine 0 REV01    740          umeshk            RE-2.0
Routing Engine 1                       270000078ba48501  RE-2.0
SSB slot 0       N/A      N/A          N/A               backup
SSB slot 1       REV 04   710-001411   AD0281            Internet Processor I
FPC 1            REV 01   710-001292   AC9230
  PIC 0          REV 01   750-001323   AA2812            1x Tunnel
  PIC 1          REV 01   750-002963   AK8586            4x T3
  PIC 2          REV 03   750-000612   AM8116            2x OC-3 ATM, MM
  PIC 3          REV 08   750-001072   AB9884            1x G/E, 1000 BASE-SX
FPC 3            REV 01   710-001197   AA8661
  PIC 2          REV 01   750-003748   HE9734            2x OC-3 SONET, SMIR

user@host> show chassis environment
Class Item                 Status     Measurement
Power Power Supply A       Absent
      Power Supply B       OK         25 degrees C / 77 degrees F
Temp  FPC 1                OK         27 degrees C / 80 degrees F
      FPC 3                OK         28 degrees C / 82 degrees F
      SSB 1                OK         29 degrees C / 84 degrees F
      Backplane            OK         23 degrees C / 73 degrees F
      Routing Engine 0     OK         25 degrees C / 77 degrees F
      Routing Engine 1     OK         24 degrees C / 75 degrees F
Fans  Rear Fan             OK         Spinning at normal speed
      Front Upper Fan      OK         Spinning at normal speed
      Front Middle Fan     OK         Spinning at normal speed
      Front Bottom Fan     OK         Spinning at normal speed
Misc  Craft Interface      OK

user@host> show chassis fpc
                    Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty             0      0          0         0         0          0
  1  Online           27      8          7         8         9         14
  2  Empty             0      0          0         0         0          0
  3  Online           28      0          0         8         8         14
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M10 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 25 degrees C / 77 degrees F
    DRAM                       768 MB
    Memory utilization           8 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     1 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                        RE-2.0
    Serial ID                    ba0000061779d601
    Start time                   2002-06-21 15:37:36 PDT
    Uptime                       2 days, 21 hours, 27 minutes, 25 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.00       0.00       0.00
Routing Engine status:
  Slot 1:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                 24 degrees C / 75 degrees F
    DRAM                       768 MB
    Memory utilization           9 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     0 percent
      Interrupt                  0 percent
      Idle                      99 percent
    Model                        RE-2.0
    Serial ID                    270000078ba48501
    Start time                   2002-06-17 14:30:21 PDT
    Uptime                       6 days, 22 hours, 34 minutes, 28 seconds
```

To verify SSB status, use the show chassis SSB command. Sample command output from an M10 router is listed below.

```
user@host> show chassis ssb
SSB status:
Slot 0 information:
  State                          Backup
Slot 1 information:
  State                          Master
  Temperature                 29 degrees C / 84 degrees F
  CPU utilization              1 percent
  Interrupt utilization        0 percent
  Heap utilization             8 percent
  Buffer utilization          43 percent
  Total CPU DRAM              64 MB
  Internet Processor I           Version 1, Foundry IBM, Part number 3
  Start time:                    2002-06-21 15:38:53 PDT
  Uptime:                        2 days, 21 hours, 26 minutes, 26 seconds
```

Table 37 provides an example of jnxFruContent objects in the jnxFruTable for an M160 router.

Table 37: jnxFruContents Objects in the jnxFruTable of an M160 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	online	0	none	0:00:00.00	0:00:12.83	6906955
2	2	0	0	PEM 1	powerEntryModule	2	online	0	none	0:00:00.00	0:00:12.83	6906955
4	1	0	0	Front Top Blower	fan	1	present	0	none	0:00:00.00	0:00:00.00	0
4	2	1	0	Fan Tray Front Left	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	2	0	Fan Tray Front Right	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	3	0	Fan Tray Rear Left	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	2	4	0	Fan Tray Rear Right	fan	2	present	0	none	0:00:00.00	0:00:00.00	0
4	3	0	0	Rear Top Blower	fan	3	present	0	none	0:00:00.00	0:00:00.00	0
4	4	0	0	Rear Bottom Blower	fan	4	present	0	none	0:00:00.00	0:00:00.00	0
6	1	1	0	SFM 0 SPP	switchingAndForwardingModule	1	online	35	none	0:00:03.13	0:00:00.00	0
6	1	2	0	SFM 0 SPR Internet Processor II	switchingAndForwardingModule	1	online	35	none	0:00:03.13	0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAndForwardingModule	2	empty	0	none	0:00:00.00	0:00:00.00	0
6	2	2	0	SFM 1 SPR	switchingAndForwardingModule	2	empty	0	none	0:00:00.00	0:00:00.00	0
6	3	1	0	SFM 2 SPP	switchingAndForwardingModule	3	online	44	none	0:00:03.20	0:00:00.00	0
6	3	2	0	SFM 2 SPR Internet Processor II	switchingAndForwardingModule	3	online	44	none	0:00:03.20	0:00:00.00	0
6	4	1	0	SFM 3 SPP	switchingAndForwardingModule	4	offline	0	configured to power off	0:00:03.22	0:00:00.00	0
6	4	2	0	SFM 3 SPR	switchingAndForwardingModule	4	offline	0	configured to power off	0:00:03.22	0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	offline	0	error	0:13:08.12	0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	online	30	none	0:00:02.32	0:00:00.00	0
7	4	0	0	FPC: 1x OC-192 SM LR @ 3/*/*	flexiblePicConcentrator	4	online	41	none	0:00:02.34	0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	empty	0	none	0:00:00.00	0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	offline	0	configured to power off	0:00:02.37	0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	7	0	0	FPC @ 6/*/*	flexiblePicConcentrator	7	empty	0	none	0:00:00.00	0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	online	41	none	0:00:03.11	0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	online	40	none	0:00:00.00	0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	2	online	46	none	0:00:00.00	0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	configured to power off	0:00:02.28	0:00:00.00	0
8	4	1	0	PIC: 1x OC-192 SM LR @ 3/0/*	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	2	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	3	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	4	4	0	PIC continued	portInterfaceCard	4	offline	0	error	0:13:08.12	0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	online	30	none	0:00:02.32	0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	online	41	none	0:00:02.34	0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard(11)	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard(11)	7	empty	0	none	0:00:00.00	0:00:00.00	0
8	8	1	0	PIC: 1x OC-12 SONET, SMIR @ 7/0/*	portInterfaceCard	8	offline	0	configured to power off	0:00:02.37	0:00:00.00	0
8	8	2	0	PIC: 4x E3 @ 7/1/*	portInterfaceCard	8	offline	0	configured to power off	0:00:02.37	0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	3	0	PIC: 1x OC-12 SONET, MM @ 7/2/* jnxFruName	portInterfaceCard	8	offline	0	configured to power off	0:00:02.37	0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	offline	0	configured to power off	0:00:02.37	0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	online	31	none	0:00:00.00	0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	2	present	0	none	0:00:00.00	0:00:00.00	0
10	1	1	0	FPM CMB	frontPanelModule	1	online	28	none	0:00:00.00	0:00:00.00	0
10	1	2	0	FPM Display	frontPanelModule	1	online	28	none	0:00:00.00	0:00:00.00	0
11	1	0	0	PCG 0	clockGenerator	1	online	40	none	0:00:00.00	0:00:00.00	0
11	2	0	0	PCG 1	clockGenerator	2	online	46	none	0:00:00.00	0:00:00.00	0
12	1	0	0	MCS 0	controlBoard	1	online	47	none	0:00:00.00	0:00:00.00	0
12	2	0	0	MCS 1	controlBoard	2	empty	0	none	0:00:00.00	0:00:00.00	0
13	1	0	0	CIP	craftInterfacePanel	1	present	0	none	0:00:00.00	0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M160 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                47                M160
Midplane         REV 02   710-001245   AB4113
FPM CMB          REV 01   710-001642   AA9721
FPM Display      REV 01   710-001647   AA2995
CIP              REV 02   710-001593   AA9886
PEM 0            Rev 01   740-001243   KJ35782           DC
PEM 1            Rev 01   740-001243   kj35756           DC
PCG 0            REV 01   710-001568   AA9796
PCG 1            REV 01   710-001568   AA9895
Routing Engine 0 REV01    740-003239   AARCHOO           RE-2.0
Routing Engine 1
MCS 0            REV 03   710-001226   AA9779
SFM 0 SPP        REV 07   710-001228   AE5504
SFM 0 SPR        REV 03   710-002189   AE4707            Internet Processor II
SFM 2 SPP        REV 06   710-001228   AB3133
SFM 2 SPR        REV 01   710-002189   AB2941            Internet Processor II
SFM 3 SPP        REV 07   710-001228   AV3167
SFM 3 SPR        REV 04   710-002189   AV3439            Internet Processor II
FPC 0            REV 02   710-001611   AA9518            FPC Type 2
  CPU            REV 02   710-001217   AA9572
FPC 1            REV 03   710-001255   AA9812            FPC Type 1
  CPU
FPC 2            REV 02   710-001611   AA9527            FPC Type 2
  CPU            REV 02   710-001217   AA9592
FPC 3            REV 01   710-003061   HB2029            FPC Type OC192
  CPU            REV 05   710-001217   AF5950
  PIC 0          REV 01   750-003063   HB2029            1x OC-192 SM LR
FPC 5            REV 01   710-001255   AA2914            FPC Type 1
  CPU            REV 02   710-001217   AA2893
```

```
FPC 7            REV 03   710-001255   AA9809           FPC Type 1
  CPU            REV 02   710-001217   AA9573
  PIC 0          REV 04   750-000613   AA0374           1x OC-12 SONET, SMIR
  PIC 1          REV 02   750-E3-PIC   AC1903           4x E3
  PIC 2          REV 02   750-001020   AA8944           1x OC-12 SONET, MM
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M160 router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Announce offline   0      0          0         0         0          0
  1  Present            0      0          0         0         0          0
  2  Online            32      4          0        32         1         39
  3  Online            44      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Offline         --- Chassis connection dropped ---
  6  Empty              0      0          0         0         0          0
  7  Online            42      4          0        32         1         40
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M160 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                       768 MB
    Memory utilization          10 percent
    CPU utilization:
      User                       1 percent
      Background                 0 percent
      Kernel                    10 percent
      Interrupt                  3 percent
      Idle                      87 percent
    Model                          RE-2.0
    Serial ID                      0c000004f8d26401
    Start time                     2002-06-14 14:39:03 PDT
    Uptime                         11 minutes, 46 seconds
    Load averages:                 1 minute   5 minute  15 minute
                                       0.18       0.19       0.14
Routing Engine status:
  Slot 1:
    Current state                  Present
```

To verify SFM status, use the show chassis sfm command. Sample command output from an M160 router is listed below.

```
user@host> show chassis sfm
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            35      1          0        64        16         46
  1  Empty              0      0          0         0         0          0
  2  Online            47      1          0        64        16         45
  3  Online            50      1          0        64        16         45

Packet scheduling mode : Disabled
```

Table 38 provides an example of jnxFruContent objects in the jnxFruTable for an M40 router.

Table 38: jnxFruContents Objects in the jnxFruTable of an M40 Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	Power Supply A	powerEntryModule	1	online	0	none	0:0:00:00.00	0:0:00:00.00	101974
2	2	0	0	Power Supply B	powerEntryModule	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
3	1	0	0	Top Impeller	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
3	2	0	0	Bottom impeller	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	0	0	Rear Left Fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	0	0	Rear Center Fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear Right Fan	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
5	1	0	0	Host controller	routingEngine	1	online	37	none	0:0:00:00.00	0:0:00:00.00	0
6	1	0	0	SCB Internet Processor I	controlBoard	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	28	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	29	none	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	24	none	0:0:00:00.00	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	online	27	none	0:0:00:00.00	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	7	0	0	FPC: 1x OC-48 SONET, SMIR @ 6/*/*	flexiblePicConcentrator	7	online	28	none	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: 1x G/E, 1000 BASE-SX @ 0/0/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x Tunnel @ 0/1/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x T1, RJ48 @ 0/2/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x COC12, SMIR @ 0/3/*	portInterfaceCard	1	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 2x OC-3 ATM, MM @ 1/0/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: 4x OC-3 SONET, MM @ 1/1/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 2x T3 @ 1/2/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x CSTM1, SMIR @ 1/3/*	portInterfaceCard	2	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	4	2	0	PIC: 4x F/E, 100 BASE-TX @ 3/1/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: 1x 800M Crypto @ 3/2/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: 1x CT3-NxDS0 @ 3/3/*	portInterfaceCard	4	ready	24	none	0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	ready	27	none	0:0:00:00.00	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: 1x OC-48 SONET, SMIR @ 6/0/*	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC continued	portInterfaceCard	7	ready	28	none	0:0:00:00.00	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	offline	0	none	0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine	routingEngine	1	online	0	none	0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M40 router is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis
Backplane        REV 03   710-000073   AA2005
Power Supply A   Rev A    740-000235   000119            DC
Maxicab          REV 04   710-000229   AA0691
Minicab          REV 02   710-000482   AA0270
Display          REV 06   710-000150   AA1042
Routing Engine                                          RE-1.0
SCB              REV 07   710-000075   AA1033            Internet Processor I
FPC 0            REV 01   710-001292   AB8159
  PIC 0          REV 08   750-001072   AP5525            1x G/E, 1000 BASE-SX
  PIC 1          REV 01   750-001323   AB1645            1x Tunnel
  PIC 2          REV 01   750-002953   AD9083            4x T1, RJ48
  PIC 3          REV 03   750-001190   AE2907            1x COC12, SMIR
FPC 1            REV 10   710-000175   AA7219
  PIC 0          REV 03   750-002977   HD9331            2x OC-3 ATM, MM
  PIC 1          REV 04   750-002971   HC8020            4x OC-3 SONET, MM
  PIC 2          REV 02.1 710-000608   AA1592            2x T3
  PIC 3          REV 05   750-003248   AD9648            1x CSTM1, SMIR
```

```
FPC 3            REV 10   710-000175   AA4782
  PIC 1          REV 04   750-002992   HC3974           4x F/E, 100 BASE-TX
  PIC 2          REV 03   750-003844   AY4806           1x 800M Crypto
  PIC 3          REV 03   750-004743   BD9433           1x CT3-NxDS0
FPC 4            REV 01   710-001292   AC5265
FPC 6            REV 01   710-001292   AB7485
  PIC 0          REV 03   750-000617   AA4566           1x OC-48 SONET, SMIR
```

```
user@host> show chassis environment
Class Item                 Status     Measurement
Power Power Supply A       OK
      Power Supply B       Absent
Temp  FPC 0                OK         28 degrees C / 82 degrees F
      FPC 1                OK         29 degrees C / 84 degrees F
      FPC 3                OK         24 degrees C / 75 degrees F
      FPC 4                OK         27 degrees C / 80 degrees F
      FPC 6                OK         28 degrees C / 82 degrees F
      SCB                  OK         27 degrees C / 80 degrees F
      Backplane @ A1       OK         30 degrees C / 86 degrees F
      Backplane @ A2       OK         26 degrees C / 78 degrees F
      Routing Engine       OK         37 degrees C / 98 degrees F
Fans  Top Impeller         OK         Spinning at normal speed
      Bottom impeller      OK         Spinning at normal speed
      Rear Left Fan        OK         Spinning at normal speed
      Rear Center Fan      OK         Spinning at normal speed
      Rear Right Fan       OK         Spinning at normal speed
Misc  Craft Interface      OK
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M40 router is listed below.

```
user@host> show chassis fpc
                    Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online           28      2          0         8        11         14
  1  Online           29      7          0         8        21         14
  2  Empty             0      0          0         0         0          0
  3  Online           24     17          0         8        22         15
  4  Online           27      1          0         8         6         13
  5  Empty             0      0          0         0         0          0
  6  Online           28      1          0         8         7         15
  7  Empty             0      0          0         0         0          0
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M40 router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                37 degrees C / 98 degrees F
    DRAM                       256 MB
    Memory utilization         19 percent
    CPU utilization:
      User                      1 percent
      Background                0 percent
      Kernel                    3 percent
      Interrupt                 1 percent
      Idle                     96 percent
    Model                      RE-1.0
    Start time                    2002-06-24 17:28:30 UTC
    Uptime                        20 minutes, 30 seconds
    Load averages:                1 minute   5 minute  15 minute
                                      0.00       0.04       0.11
```

To verify SCB status, use the show chassis scb command. Sample command output from an M40 router is listed below.

```
user@host> show chassis scb
SCB status:
  Temperature                   27 degrees C / 80 degrees F
  CPU utilization               3 percent
  Interrupt utilization         0 percent
  Heap utilization              9 percent
  Buffer utilization            44 percent
  Total CPU DRAM                64 MB
  Internet Processor I          Version 1, Foundry IBM, Part number 3
  Start time:                   2002-06-24 17:30:10 UTC
  Uptime:                       19 minutes, 8 seconds
```

Table 39 provides an example of JnxFruContent objects in the JnxFruTable for an M40e router.

Table 39: JnxFruContents Objects in the jnxFruTable of an M40e Router

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	present	0	none	0:0:00:00.00	0:0:00:25.99	208927
2	2		0	PEM 1	powerEntryModule	2	online	0	none	0:0:00:00.00	0:0:00:25.99	208928
4	1	0	0	Front Top Blower	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Fan Tray Front Left	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Fan Tray Front Right	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Fan Tray Rear Left	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Fan Tray Rear Right	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	0	0	Rear Top Blower	fan	3	present		none	0:0:00:00.00	0:0:00:00.00	0
4	4	0	0	Rear Bottom Blower	fan	4	present	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	1	0	SFM 0 SPP	switchingAndForwardingModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
6	1	2	0	SFM 0 SPR	switchingAndForwardingModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
6	2	1	0	SFM 1 SPP	switchingAndForwardingModule	2	online	42	none	0:0:00:21.95	0:0:00:00.00	0
6	2	2	0	SFM 1 SPR Internet Processor II	switchingAndForwardingModule	2	online	42	none	0:0:00:21.95	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	online	41	none	0:0:00:21.85	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	online	43	none	0:0:00:21.87	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	online	38	none	0:0:00:21.89	0:0:00:00.00	0
7	5	0	0	FPC @ 4/*/*	flexiblePicConcentrator	5	empty	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	online	46	none	0:0:00:21.91	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePicConcentrator	7	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	online	44	none	0:0:00:21.93	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: 1x OC-12 SONET, MM @ 0/1/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: 4x CT3 @ 0/2/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: 1x Multi Link(32) @ 0/3/*	portInterfaceCard	1	online	45	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: @ 1/0/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	2	0	PIC: @ 1/1/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: @ 1/2/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: @ 1/3/*	portInterfaceCard	2	online	50	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: 1x OC-12 SONET, MM @ 2/0/*	portInterfaceCard	3	online	41	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: 1x OC-12 SONET, MM @ 2/1/*	portInterfaceCard	3	online	41	none	0:0:00:21.85	0:0:00:00.00	0
8	3	3	0	PIC: 1x OC-12 SONET, MM @ 2/2/*	portInterfaceCard	3	online	41		0:0:00:21.85	0:0:00:00.00	
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	online	41		0:0:00:21.85	0:0:00:00.00	
8	4	1	0	PIC: 1x OC-48 SONET, SMIR @ 3/0/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	5	1	0	PIC: @ 4/0/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	2	0	PIC: @ 4/1/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	3	0	PIC: @ 4/2/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	5	4	0	PIC: @ 4/3/*	portInterfaceCard	5	online	43		0:0:00:21.87	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	3	0	PIC: 1x OC-12 SONET, SMIR @ 5/2/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	6	4	0	PIC: 1x OC-12 SONET, MM @ 5/3/*	portInterfaceCard	6	online	38		0:0:00:21.89	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard	7	empty	0		0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	8	1	0	PIC: 8x FE-FX, 100 BASE-FX @ 7/0/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	3	0	PIC: 1x Link Service(4) @ 7/2/*	portInterfaceCard	8	online	46		0:0:00:21.91	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	1	online	46		0:0:00:00.00	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	2	online	46		0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	1	present	34		0:0:00:00.00	0:0:00:00.00	0
10	1	1	0	FPM CMB	frontPanelModule	1	online	28		0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanelModule	1	online	28		0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	PCG 0	clockGenerator	1	online	45		0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	PCG 1	clockGenerator	2	online	50		0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	MCS 0	controlBoard	1	online	46		0:0:00:00.00	0:0:00:00.00	0
12	2	0	0	MCS 1	controlBoard	2	online	0		0:0:00:00.00	0:0:00:00.00	0
13	1	0	0	CIP	craftInterfacePanel	1	present	0		0:0:00:00.00	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from an M40e router is listed below.

```
user@host> show chassis hardware
Item             Version  Part number  Serial number     Description
Chassis                               19084             M40e
Midplane         REV 01   710-005071   AX3654
FPM CMB          REV 03   710-001642   AR9037
FPM Display      REV 03   710-001647   AP1334
CIP              REV 08   710-001593   AE8486
PEM 0            Rev 01   740-003787   ME13120           Power Entry Module
PEM 1            Rev 01   740-003787   MC25354           Power Entry Module
PCG 0            REV 07   710-001568   AG1377
PCG 1            REV 07   710-001568   AR3806
Routing Engine 0 REV 04   740-003239   9001026568        RE-2.0
Routing Engine 1
MCS 0            REV 11   710-001226   AN5810
MCS 1            REV 11   710-001226   AR0109
SFM 1 SPP        REV 07   710-001228   BE0106
SFM 1 SPR        REV 05   710-002189   BE0062            Internet Processor II
FPC 0            REV 01   710-005078   BE0642            M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2496
  PIC 1          REV 04   750-001895   HE0885            1x OC-12 SONET, MM
  PIC 2          REV 06   750-003009   HE1422            4x CT3
  PIC 3          REV 03   750-003837   AP7134            1x Multi Link(32)
FPC 2            REV 01   710-005078   BE0647            M40e-FPC Type 1
  CPU            REV 01   710-004600   AN4299
  PIC 0          REV 04   750-001895   HD2623            1x OC-12 SONET, MM
  PIC 1          REV 04   750-001895   HE0609            1x OC-12 SONET, MM
  PIC 2          REV 04   750-001895   HE0871            1x OC-12 SONET, MM
FPC 3            REV 01   710-005197   BD9846            M40e-FPC Type 2
  CPU            REV 01   710-004600   BD2364
  PIC 0          REV 01   750-001900   AA9649            1x OC-48 SONET, SMIR
```

```
FPC 5            REV 01   710-005078   BE0639           M40e-FPC Type 1
  CPU            REV 01   710-004600   BD2587
  PIC 2          REV 04   750-001896   AV4480           1x OC-12 SONET, SMIR
  PIC 3          REV 04   750-001895   HE1000           1x OC-12 SONET, MM
FPC 7            REV 01   710-005196   BD9456           M40e-FPC
  CPU            REV 01   710-004600   AN4323
  PIC 0          REV 01   750-004944   AY4645           8x FE-FX, 100 BASE-FX
  PIC 2          REV 01   750-007927   AP1919           1x Link Service(4)
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output from an M40e router is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state               Master
    Election priority           Master (default)
    Temperature                  34 degrees C / 93 degrees F
    DRAM                        768 MB
    Memory utilization          9 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    2 percent
      Interrupt                 0 percent
      Idle                     97 percent
    Model                       RE-2.0
    Serial ID                   9c000007c8644701
    Start time                  2002-06-24 10:33:41 PDT
    Uptime                      31 minutes, 7 seconds
    Load averages:              1 minute   5 minute  15 minute
                                    0.01       0.02       0.00
Routing Engine status:
  Slot 1:
    Current state               Present
```

To verify FPC status, use the show chassis fpc command. Sample command output from an M40e router is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online            41      4          0        32         3         40
  1  Empty              0      0          0         0         0          0
  2  Online            43      4          0        32         1         40
  3  Online            38      1          0        32         1         40
  4  Empty              0      0          0         0         0          0
  5  Online            46      4          0        32         1         40
  6  Empty              0      0          0         0         0          0
  7  Online            44      4          0        32         2         39
```

Table 40 provides an example of jnxFruContent objects in the jnxFruTable for a T640 Routing Node.

Table 40: jnxFruContents Objects in the jnxFruTable of a T640 Routing Node

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
2	1	0	0	PEM 0	powerEntryModule	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
2	2	0	0	PEM 1	powerEntryModule	2	online	27	none	0:0:00:00.00	0:0:00:00.00	217044
4	1	1	0	Top Left Front fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	2	0	Top Left Middle fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	3	0	Top Left Rear fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	4	0	Top Right Front fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	5	0	Top Right Middle fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	1	6	0	Top Right Rear fan	fan	1	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	1	0	Bottom Left Front fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	2	0	Bottom Left Middle fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	3	0	Bottom Left Rear fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	4	0	Bottom Right Front fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	5	0	Bottom Right Middle fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	2	6	0	Bottom Right Rear fan	fan	2	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	1	0	Fourth Blower from top	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	2	0	Bottom Blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	3	0	Middle Blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	4	0	Top Blower	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
4	3	5	0	Second Blower from top	fan	3	present	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	0	0	FPC @ 0/*/*	flexiblePicConcentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	1	0	FPC @ 0/0/* top temp sensor	flexiblePicConcentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	1	2	0	FPC @ 0/1/* bottom temp sensor	flexiblePicConcentrator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	2	0	0	FPC @ 1/*/*	flexiblePicConcentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0
7	2	1	0	FPC @ 1/0/* top temp sensor	flexiblePicConcentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0
7	2	2	0	FPC @ 1/1/* bottom temp sensor	flexiblePicConcentrator	2	online	30	none	0:0:00:01.94	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
7	3	0	0	FPC @ 2/*/*	flexiblePicConcentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	3	1	0	FPC @ 2/0/* top temp sensor	flexiblePicConcentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	3	2	0	FPC @ 2/1/* bottom temp sensor	flexiblePicConcentrator	3	online	30	none	0:0:00:01.96	0:0:00:00.00	0
7	4	0	0	FPC @ 3/*/*	flexiblePicConcentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	1	0	FPC @ 3/0/* top temp sensor	flexiblePicConcentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	4	2	0	FPC @ 3/1/* bottom temp sensor	flexiblePicConcentrator	4	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	5	9	0	FPC @ 4/*/*	flexiblePicConcentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0
7	5	1	0	FPC @ 4/0/* top temp sensor	flexiblePicConcentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0
7	5	2	0	FPC @ 4/1/* bottom temp sensor	flexiblePicConcentrator	5	online	36	none	0:0:00:01.98	0:0:00:00.00	0
7	6	0	0	FPC @ 5/*/*	flexiblePicConcentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	6	1	0	FPC @ 5/0/* top temp sensor	flexiblePicConcentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	6	2	0	FPC @ 5/1/* bottom temp sensor	flexiblePicConcentrator	6	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
7	7	0	0	FPC @ 6/*/*	flexiblePicConcentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	7	1	0	FPC @ 6/0/* top temp sensor	flexiblePicConcentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	7	2	0	FPC @ 6/1/* bottom temp sensor	flexiblePicConcentrator	7	online	30	none	0:0:00:02.05	0:0:00:00.00	0
7	8	0	0	FPC @ 7/*/*	flexiblePicConcentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	8	1	0	FPC @ 7/0/* top temp sensor	flexiblePicConcentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
7	8	2	0	FPC @ 7/1/* bottom temp sensor	flexiblePicConcentrator	8	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	1	0	PIC: @ 0/0/*	portInterfaceCard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	2	0	PIC: @ 0/1/*	portInterfaceCard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	3	0	PIC: @ 0/2/*	portInterfaceCard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	1	4	0	PIC: @ 0/3/*	portInterfaceCard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	2	1	0	PIC: 1x OC-48 SONET, SMIR @ 1/0/*	portInterfaceCard	2	online		none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
8	2	2	0	PIC: 1x OC-48 SONET, SMSR @ 1/1/*	portInterfaceCard	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	2	3	0	PIC: 1x OC-48 SONET, SMIR @ 1/2/*	portInterfaceCard	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	2	4	0	PIC: 1x OC-48 SONET, SMIR @ 1/3/*	portInterfaceCard	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
8	3	1	0	PIC: @ 2/0/*	portInterfaceCard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	2	0	PIC: @ 2/1/*	portInterfaceCard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	3	0	PIC: @ 2/2/*	portInterfaceCard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	3	4	0	PIC: @ 2/3/*	portInterfaceCard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	4	1	0	PIC: @ 3/0/*	portInterfaceCard	4	online		none	0:0:00:01.00	0:0:00:00.00	0
8	4	2	0	PIC: @ 3/1/*	portInterfaceCard	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	4	3	0	PIC: @ 3/2/*	portInterfaceCard	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	4	4	0	PIC: @ 3/3/*	portInterfaceCard	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	5	1	0	PIC: 1x Tunnel @ 4/0/*	portInterfaceCard	5	online	30	none	0:0:00:01.94	0:0:00:00.00	0
8	5	2	0	PIC: 1x OC-192 SM SR2 @ 4/1/*	portInterfaceCard	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0
8	5	3	0	PIC: 4x OC-48 SONET, SMSR @ 4/2/*	portInterfaceCard	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0
8	5	4	0	PIC: 1x OC-192 SM SR1 @ 4/3/*	portInterfaceCard	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0
8	6	1	0	PIC: @ 5/0/*	portInterfaceCard	6	empty	0	none	0:0:00:01.00	0:0:00:00.00	0
8	6	2	0	PIC: @ 5/1/*	portInterfaceCard	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	3	0	PIC: @ 5/2/*	portInterfaceCard	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	6	4	0	PIC: @ 5/3/*	portInterfaceCard	6	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
8	7	1	0	PIC: @ 6/0/*	portInterfaceCard	7	online	30	none	0:0:00:00.00	0:0:00:00.00	0
8	7	2	0	PIC: @ 6/1/*	portInterfaceCard	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	7	3	0	PIC: @ 6/2/*	portInterfaceCard	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	7	4	0	PIC: @ 6/3/*	portInterfaceCard	7	online	30	none	0:0:00:01.98	0:0:00:00.00	0
8	8	1	0	PIC: @ 7/0/*	portInterfaceCard	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	2	0	PIC: @ 7/1/*	portInterfaceCard	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	3	0	PIC: @ 7/2/*	portInterfaceCard	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
8	8	4	0	PIC: @ 7/3/*	portInterfaceCard	8	offline	0	error	0:0:12:51.28	0:0:00:00.00	0
9	1	0	0	Routing Engine 0	routingEngine	1	online	34	none	0:0:00:00.00	0:0:00:00.00	0
9	2	0	0	Routing Engine 1	routingEngine	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0

Contents index	L1	L2	L3	Name	Type	Slot	State	Temp	Offline	PowerOff	PowerOn	Uptime
10	1	1	0	FPM GBUS	frontPanelModule	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
10	1	2	0	FPM Display	frontPanelModule	1	online	27	none	0:0:00:00.00	0:0:00:00.00	0
11	1	0	0	SCG 0	clockGenerator	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
11	2	0	0	SCG 1	clockGenerator	2	online	27	none	0:0:00:00.00	0:0:00:00.00	0
12	1	0	0	CB 0	controlBoard	1	online	27	none	0:0:00:01.94	0:0:00:00.00	0
12	2	0	0	CB 1	controlBoard	2	unknown	0	none	0:0:00:01.96	0:0:00:00.00	0
13	1	0	0	CIP	craftInterfacePanel	1	present	36	none	0:0:00:00.00	0:0:00:00.00	0
14	1	0	0	SPMB 0	processorMezzanineBoardForSIB	1	online	34	none	0:0:00:00.00	0:0:00:00.00	0
14	2	0	0	SPMB 1	processorMezzanineBoardForSIB	2	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	1	0	0	SIB 0	switchInterfaceBoard	1	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	2	0	0	SIB 1	switchInterfaceBoard	2	online	36	none	0:0:00:00.00	0:0:00:00.00	0
15	3	0	0	SIB 2	switchInterfaceBoard	3	empty	0	none	0:0:00:00.00	0:0:00:00.00	0
15	4	0	0	SIB 3	switchInterfaceBoard	4	online	30	none	0:0:00:01.94	0:0:00:00.00	0
15	5	0	0	SIB 4	switchInterfaceBoard	5	online	30	none	0:0:00:01.96	0:0:00:00.00	0

To verify the L1, L2, and L3 indexes, use the show chassis hardware command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                1234              T640
Midplane         REV 04   710-002726   AX5603
FPM GBUS         REV 02   710-002901   HE3062
FPM Display      REV 01   710-002897   HD3033
CIP              REV 05   710-002895   HA5022
PEM 1            RevX02   740-002595   MD21812           Power Entry Module
SCG 1            REV 01   710-003423   HD3025
Routing Engine 0 REV 01   740-005022   210865700336      RE-3.0
CB 0             REV 02   710-002728   HE3025
CB 1
FPC 1            REV 01   710-002385   HE3173            FPC Type 2
  CPU            REV 06   710-001726   HC0042
  PIC 0          REV 03   750-001900   AD5737            1x OC-48 SONET, SMIR
  PIC 1          REV 07   750-001900   AR3613            1x OC-48 SONET, SMSR
  PIC 2          REV 01   750-001900   AA9604            1x OC-48 SONET, SMIR
  PIC 3          REV 01   750-001900   AA9602            1x OC-48 SONET, SMIR
  MMB 1          REV 03   710-001723   HC0111            MMB-144mbit
  ICBM           REV 04   710-003384   HA4497
  PPB 0          REV 02   710-003758   HA4543            PPB Type 2
  PPB 1          REV 02   710-003758   HA4540            PPB Type 2
FPC 2            REV 01   710-002385   HE3180            FPC Type 2
  CPU            REV 06   710-001726   HE7904
  MMB 1          REV 03   710-001723   HC0120            MMB-144mbit
  ICBM           REV 01   710-003384   HE3046
  PPB 0          REV 02   710-003758   HA4564            PPB Type 2
  PPB 1          REV 02   710-003758   HA4554            PPB Type 2
```

```
FPC 4            REV 04   710-001721   HE3145           FPC Type 3
  CPU            REV 06   710-001726   HC0034
  PIC 0                                                  1x Tunnel
  PIC 1          REV 01   750-003824   HE7803           1x OC-192 SM SR2
  PIC 2          REV 01   750-003336   HE3420           4x OC-48 SONET, SMSR
  PIC 3          REV 01   750-003824   HE7802           1x OC-192 SM SR1
  MMB 0          REV 03   710-001723   HE7230           MMB-144mbit
  MMB 1          REV 03   710-001723   HE7267           MMB-144mbit
  ICBM           REV 04   710-003384   HA4485
  PPB 0          REV 02   710-002845   HA4550           PPB Type 3
  PPB 1          REV 02   710-002845   HA4525           PPB Type 3
FPC 5            REV 04   710-001721   HE3175           FPC Type 3
  CPU
FPC 6            REV 01   710-002385   HD5027           FPC Type 2
  CPU            REV 06   710-001726   HC0033
  MMB 1          REV 03   710-001723   HC0080           MMB-144mbit
  ICBM           REV 04   710-003384   HA4486
  PPB 0          REV 02   710-003758   HA4541           PPB Type 2
  PPB 1          REV 02   710-003758   HA4539           PPB Type 2
SPMB 0           REV 01   710-003229   HA5999
SIB 0            REV 01   710-003980   HD5054           SIB-I8
SIB 2            REV 01   710-003980   HC0035           SIB-I8
SIB 3            REV 01   710-003980   HA5065           SIB-I8
SIB 4            REV 01   710-003980   HE3016           SIB-I8
```

To verify FPC status, use the show chassis fpc command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis fpc
Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Empty              0      0          0         0         0          0
  1  Online            30      2          0       512         3         41
  2  Online            30      2          0       256         7         41
  3  Empty              0      0          0         0         0          0
  4  Online            30      4          0       512         6         41
  5  Offline         --- Unresponsive ---
  6  Online            30      2          0       256         7         41
  7  Empty              0      0          0         0         0          0
```

To verify Routing Engine status, use the show chassis routing-engine command. Sample command output a T640 routing node is listed below.

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 35 degrees C / 95 degrees F
    DRAM                      2048 MB
    Memory utilization           4 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     2 percent
      Interrupt                  0 percent
      Idle                      97 percent
    Model                        RE-3.0
    Start time                   2002-06-24 10:33:34 PDT
    Uptime                       33 minutes, 38 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.08       0.03       0.01
```

To verify SPMB status, use the show chassis spmb command. Sample command output from a T640 routing node is listed below.

```
user@host> show chassis spmb
Slot 0 information:
  State                          Online
  Total CPU Utilization             2%
  Interrupt CPU Utilization         0%
  Memory Heap Utilization           0%
  Buffer Utilization               40%
  Start time:                    2002-06-24 10:34:22 PDT
  Uptime:                         33 minutes, 3 seconds
```

jnxTraps

The chassis-related traps are defined under the jnxTraps branch.For the system logging severity levels for these traps, see “Juniper Networks Enterprise-Specific SNMP Traps” on page 59.

These traps are defined as follows:

- Power failure (jnxPowerSupplyFailure)—When the power supply, router circuit breaker, or power circuit fails, or when there is a power outage. If only one of the power supplies in the router fails, the service impact is minimal. One power supply can provide the necessary power for a fully loaded router. To determine the source of the failure, you must physically inspect the router.

- Fan failure (jnxFanFailure)—When the fan fuse blows or when the fan wiring shorts out. If only one fan has failed, there is no service impact. The remaining fans increase speed to compensate. However, you must resolve the problem before another fan fails. To determine the source of the failure, you must physically inspect the router, taking care to check the fuses. See the hardware installation guide for your router model for more information.

- Overtemperature (jnxOverTemperature)—When several fans fail or the room temperature increases significantly. The service impact of this trap depends on the temperature of the router. In general, the router increases the speed of the fans when any component exceeds a temperature of 55 °C. The fans remain at the higher speed until the temperature decreases below the threshold. In this case, there is no service impact. However, if the temperature exceeds 75 °C, the router transmits a warning and automatically shuts down. This scenario creates a significant service impact because the shutdown affects additional routers and equipment. To determine the source of the overtemperature problem, you must physically inspect the router to determine whether any fans have failed in the router.

- Power Supply OK (jnxPowerSupplyOK)—Sent when a power supply recovers from failure.

- Fan OK (jnxFanOK)—Sent when a fan recovers from failure.

- Temperature OK (jnxTemperatureOK)—Sent when a chassis component recovers from an overtemperature condition.

- Redundancy Switchover (jnxRedundancySwitchover)—For certain platforms, such as the M20 or M160, some subsystems such as the Routing Engine have a redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions. The redundancy switchover trap indicates such a change.

- Field Replaceable Unit Removal (jnxFruRemoval)—Sent when the specified FRU has been removed from the chassis.

- Field Replaceable Unit Insertion (jnxFruInsertion)—Sent when the specified FRU has been inserted into the chassis.

- Field Replaceable Unit Power Off (jnxFruIPoweroff)—Sent when the specified FRU has been powered off in the chassis.

- Field Replaceable Unit Power On (jnxFruPowerOn)—Sent when the specified FRU has been powered on in the chassis.

For more information on chassis MIB traps, see “Standard SNMP Traps” on page 77 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 59.

This section contains the following topics:

SNMPv1 Trap Format

The SNMPv1 trap format for the chassis-related traps is described in Table 41. To view the SNMPv1 chassis-related traps, see “Standard SNMP Traps” on page 77 and “Juniper Networks Enterprise-Specific SNMP Traps” on page 59.

The column headings describe the SNMPv1 traps format:

- Trap Name—The name of the trap.
- Enterprise ID—The identification number of the enterprise-specific trap.
- Generic Trap Number—The generic trap number field of the SNMP trap PDU. This field is enterpriseSpecific(6) for enterprise-specific traps, other predefined values for standard traps.
- Specific Trap Number— The specific trap number field of the SNMP trap PDU. For standard traps, this field is zero; for enterprise-specific traps, this field is nonzero as defined in the enterprise-specific MIBs.

Table 41: SNMP Version 1 Trap Format

Trap Name	Enterprise ID	Generic Trap Number	Specific Trap Number
jnxFanFailure	1.3.6.1.4.1.2636.4.1	6	2
jnxFanOK	1.3.6.1.4.1.2636.4.2	6	2
jnxFruInsertion	1.3.6.1.4.1.2636.4.1	6	6
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1	6	7
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1	6	8
jnxFruRemoval	1.3.6.1.4.1.2636.4.1	6	5
jnxOverTemperature	1.3.6.1.4.1.2636.4.1	6	3
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1	6	1
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2	6	1
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1	6	4
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2	6	3

SNMPv2 Trap Format

The SNMPv2 trap format for the chassis MIB traps is described in Table 42.

The column headings describe the SNMPv2 traps format:

- Trap Name—The name of the trap.
- snmpTrapOID—The authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2 trap PDU and InformRequest PDU.
- Description—The JUNOS enterprise-specific name of the trap.

Table 42: SNMP Version 2 Trap Format

Trap Name	snmpTrapOID	Description
jnxFanFailure	1.3.6.1.4.1.2636.4.1.2	The fan fuse blows or the fan wiring shorts out.
jnxFanOK	1.3.6.1.4.1.2636.4.2.2	The fan has recovered from failure state.
jnxFruInsertion	1.3.6.1.4.1.2636.4.1.6	The FRU has been inserted into the chassis.
jnxFruPowerOff	1.3.6.1.4.1.2636.4.1.7	The FRU has been powered off in the chassis.
jnxFruPowerOn	1.3.6.1.4.1.2636.4.1.8	The FRU has been powered on in the chassis.
jnxFruRemoval	1.3.6.1.4.1.2636.4.1.5	The FRU has been removed from the chassis.
jnxOverTemperature	1.3.6.1.4.1.2636.4.1.3	Several fans fail or the room temperature increases significantly.
jnxPowerSupplyFailure	1.3.6.1.4.1.2636.4.1.1	The power supply, router circuit breaker, or power circuit fails, or there is a power outage.
jnxPowerSupplyOK	1.3.6.1.4.1.2636.4.2.1	The power supply has recovered from failure state.
jnxRedundancySwitchover	1.3.6.1.4.1.2636.4.1.4	A redundant backup unit that can be brought online, manually or automatically, if the main unit malfunctions.
jnxTemperatureOK	1.3.6.1.4.1.2636.4.2.3	The component sensor has detected the overtemperature condition.

Chapter 14
Interpret the Destination Class Usage MIB

Destination class usage (DCU) counts packets from customers by performing a lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

The DCU MIB is a subbranch of the jnxMibs branch of the enterprise-specific MIB {enterprise 2636} and has an object identifier of {jnxMIB 6}. The DCU MIB has one branch, jnxDCUs, which contains two tables: jnxDCUsTable and jnxDcuStatsTable. For information about configuring source and destination class usage, see the *JUNOS Internet Software Configuration Guide: Policy Framework* and *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-dcu.txt.

This chapter contains the following topics:

jnxDCUsTable

The entries in the jnxDCUsTable, whose object identifier is {jnxDCUTable 1}, are represented by jnxDCUsEntry and are listed in the following table:

Table 43: jnxDCUsEntry

Object	Object Identifier	Description
jnxDCUSrcIfIndex	jnxDUCsEntry 1	The interface index of the ingress interface
jnxDCUDstClassName	jnxDCUsEntry 2	The destination class name specified in a routing policy and applied to the forwarding table.
jnxDCUPackets	jnxDCUsEntry 3	The number of packets passing through the network.
jnxDCUBytes	jnxDCUsEntry 4	The number of bytes passing through the network.

jnxDcuStatsTable

jnxDcuStatsTable contains statistics for traffic that satisfies the rules in each configured destination class. A separate set of statistics is kept for each destination class on each interface and address family on which this feature is enabled. This is essentially a replacement for jnxDCUsTable.

The entries in the jnxDcuStatsTable, whose object identifier is {jnxDCUs 2}, are represented by jnxDCUsStatusEntry and are listed in the following table:

Table 44: jnxDCUsStatusEntry

Ojbect	Object Identifier	Description
jnxDcuStatsSrcIfIndex	jnxDcuStatsEntry 1	The interface index of the ingress interface for traffic counted in each entry.
jnxDcuStatsAddrFamily	jnxDcuStatsEntry 2	The address family of the entry's traffic.
nxDcuStatsClassName	jnxDcuStatsEntry 3	The name of the destination class that applies to the entry's traffic.
jnxDcuStatsPackets	jnxDcuStatsEntry 4	The number of packets received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsBytes	jnxDcuStatsEntry 5	The number of bytes received on this interface and belonging to this address family that match this destination class.
jnxDcuStatsClName	jnxDcuStatsEntry 6	The name of the destination class. This object is a duplicate of jnxDcuStatsClassName and is included to satisfy those network management applications that cannot extract the destination class name from the instance portion of the OID.

Chapter 15
Interpret the Ping MIB

The ping MIB extends the standard Ping MIB control table (RFC 2925). Items in this MIB are created when entries are created in the pingCtlTable of the ping MIB. Each item is indexed exactly as it is in the ping MIB.

To view a complete copy of the enterprise-specific extensions to the ping MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ping.txt. For more information on using the ping MIB and enterprise-specific ping MIB, see "SNMP Remote Operations" on page 35.

jnxPingCtlTable

The enterprise-specific ping MIB structure includes one main object, jnxPingCtlTable. jnxPingCtlTable, whose object identifier is {jnxPingObjects 2}, defines the jnxPing control table for providing enterprise-specific options to the corresponding pingCtlEntry.

jnxPingCtlEntry

Each jnxPingCtlEntry has two indexes identical to those of the corresponding pingCtlEntry. Entries created in pingCtlTable are mirrored here. jnxPingCtlEntry objects are listed in the following table:

Table 45: jnxPingCtlEntry

Object	Object Identifier	Description
jnxCtlOwnerIndex	jnxPingCtlEntry 1	The first index. It is identical to the pingCtlOwnerIndex of the corresponding pingCtlEntry in the pingCtlTable.
jnxPingCtlTestName	jnxPingCtlEntry 2	The other index and is identical to the pingCtlTestName of the corresponding pingCtlEntry in the pingCtlTable.
jnxPingCtlIfName	jnxPingCtlEntry 3	Specifies the name of the outgoing interface for ping probes. This is the name-based complement to pingCtlIfIndex. A zero length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: ■ pingCtlIfIndex (see pingCtlTable in the ping MIB) ■ jnxPingCtlIfName ■ jnxPingCtlRoutingInstanceName
jnxPingCtlRoutingInstance Name	jnxPingCtlEntry 6	Specifies the name of the routing instance used when directing outgoing ping packets. The instance name specified must be configured at the [edit routing-instances] hierarchy level of the JUNOS configuration. The instance-type must be vrf.

Chapter 16
Interpret the Traceroute MIB

The traceroute MIB supports the JUNOS extensions of traceroutes and remote operations. Items in this MIB are created when entries are created in the traceRouteCtlTable of the traceroute MIB. Each item is indexed exactly the same way as it is in the traceroute MIB.

To view a complete copy of the enterprise-specific extensions to the traceroute MIB, see see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-traceroute.txt.

For more information on using the traceroute MIB and enterprise-specific traceroute MIB, see "SNMP Remote Operations" on page 35.

The enterprise-specific trcerouteMIB structure includes one main object, jnxTraceRouteCtlTable.

jnxTraceRouteCtlTable

The jnxTraceRouteCtlTable, whose object identifier is {jnxTraceRouteObjects 2}, defines the jnxTraceRoute control table for providing enterprise-specific options to the corresponding traceRouteCtlEntry.

jnxTraceRouteCtlEntry

Each jnxTraceRouteCtlEntry has two indexes identical to those of the corresponding TraceRouteCtlEntry. Entries created in TraceRouteCtlTable are mirrored here and and are listed in the following table:

Object	Object Identifier	Description
jnxTRCtlOwnerIndex	jnxTraceRouteCtlEntry 1	The first index. It is identical to the jnxTraceRouteCtlOwnerIndex of the corresponding jnxTraceRouteCtlEntry in the jnxTraceRouteCtlTable.
jnxTRCtlTestName	jnxTraceRouteCtlEntry 2	The other index and is identical to the jnxTraceRouteCtlTestName of the corresponding jnxTraceRouteCtlEntry in the jnxTraceRouteCtlTable.

Object	Object Identifier	Description
jnxTRCtlIfName	jnxTraceRouteCtlEntry 3	Specifies the name of the outgoing interface for traceroute probes. This is the name-based complement to traceRouteCtlIfIndex. A zero length string value for this object means that this option is not enabled. The following values can be set simultaneously, but only one value is used. The precedence order is as follows: ■ traceRouteCtlIfIndex (see traceRouteCtlTable in the traceroute MIB) ■ jnxTRCtlIfName ■ jnxTRCRoutingInstanceName
jnxTRCtlRoutingInstanceName	jnxTraceRouteCtlEntry 4	Specifies the name of the routing instance used when directing outgoing traceroute packets. The instance name specified must be configured at the [edit routing-instances] hierarchy level of the JUNOS configuration

Chapter 17

Interpret the RMON Events and Alarms MIB

The remote monitoring (RMON) events and alarms MIB monitors objects on a device and warns the network system administrator if one of those values exceeds the defined range. The alarm monitors objects in this MIB and triggers an event when the condition (falling or rising threshold) is reached.

The Juniper Networks enterprise-specific extension to the standard RMON MIB augments the alarmTable with additional information about each alarm. Two new traps, jnxRmonAlarmGetFailure and jnxRmonGetOk, are also defined to indicate when problems are encountered with an alarm.

To view a complete copy of the enterprise-specific extensions to the RMON MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-ldp.txt. For more information on RMON alarms and events, see "RMON Alarms and Events" on page 113.

This chapter contains the following topics:

jnxRmonAlarmTable

The entries in the jnxRmonAlarmTable, whose object identifier is {jnxMibs 13}, are represented by jnxRmonAlarmEntry, whose object identifier is { jnxRmonAlarmTable1 } and are listed in the following table:

Table 46: jnxRmonAlarmEntry

Object	Object Identifier	Description
jnxRmonAlarmGetFailCnt	jnxRmonAlarmEntry 1	Represents the number of times the internal Get request for the variable monitored by this entry has failed.
jnxRmonAlarmGetFailTime	jnxRmonAlarmEntry 2	Represents the value of sysUpTime when an internal Get request for the variable monitored by this entry last failed.

Object	Object Identifier	Description
jnxRmonAlarmGetFailReason	jnxRmonAlarmEntry 3	Represents the reason an internal Get request for the variable monitored by this entry last failed. This object contains the following values: ■ other (1)—An error was encountered that does not fit into one of the currently defined categories. ■ noError (2)—Get request processed successfully. ■ noSuchObject (3)—Requested object not available. ■ outOfView (4)—Requested object instance out of MIB view. ■ noSuchInstance (5) —Requested object instance not available. ■ badReqId (6)—Unexpected request ID encountered while processing Get request. ■ oidMatchErr (7)—Unexpected object ID encountered while processing Get request. ■ oidBindErr (8)—Unable to bind object ID to Get request PDU. ■ createPktErr (9)—Unable to create Get request PDU. ■ badObjType (10)—Unexpected object type encountered while processing Get request
jnxRmonAlarmGetOkTime	jnxRmonAlarmEntry 4	Represents the value of sysUpTime when an internal Get request for the variable monitored by this entry succeeded and the entry left the getFailure state.
jnxRmonAlarmState	jnxRmonAlarmEntry 5	Represents the current state of this RMON alarm entry. This object contains the following values: unknown (1)—Alarm entry unknown underCreation (2)—Alarm entry not activated active (3)—Alarm entry active and within thresholds startup (4)—Alarm entry still waiting for first value risingThreshold (5)—Alarm entry has crossed the rising threshold. fallingThreshold (6)—Alarm entry has crossed the falling threshold getFailure (7)—Alarm entry internal Get request failed.

RMON Event and Alarm Traps

The following traps send notifications when there is a problem with RMON alarm processing

Table 47: RMON Event and Alarm Traps

Trap	Object Identifer	Description
jnxRmonAlarmGetFailure	jnxRmonTrapPrefix 1	Generated when the Get request for an alarm variable returns an error. The specific error is identified by jnxRmonAlarmGetFailReason.
jnxRmonGetOk	jnxRmonTrapPrefix 2	Generated when the Get request for an alarm variable is successful. This trap is only sent after previous attempts are unsuccessful.

Chapter 18
Interpret the Reverse Path Forwarding MIB

The reverse path forwarding MIB monitors statistics for traffic that is rejected because of reverse path forwarding (RPF) processing. The reverse path forwarding MIB includes one main object, jnxRpfStats, with an object identifier of {jnxRpf 1}. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-rpf.txt.

This chapter discusses the following topic:

jnxRpfStatsTable

The jnxRpfStatsTable, whose object identifier is {jnxRpfStats 1}, provides a list of RPF entries in table format.

jnxRpfStatsEntry

The jnxRpfStatsEntry, whose object identifier is {jnxRpfStatsTable 1}, has four objects, which are listed in the following table:

Table 48: jnxRpfStatsEntry

Object	Object Identifier	Description
jnxRpfStatsIfIndex	jnxRpfStatsEntry 1	The ingress interface for traffic that is counted in an RpfStats entry.
jnxRpfStatsAddrFamily	jnxRpfStatsEntry 2	The address family of an entry's traffic, which can be in IPv4 or IPv6 format
jnxRpfStatsPackets	jnxRpfStatsEntry 3	The number of packets received on this interface, belonging to this address family, that have been rejected due to RPF processing.
jnxRpfStatsBytes	jnxRpfStatsEntry 4	The number of bytes received on this interface, belonging to this address family, that have been rejected due to RPF processing.

Chapter 19
Interpret the Source Class Usage MIB

The source class usage (SCU) MIB counts packets sent to customers by performing a lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge.

The SCU MIB is an object of the jnxMibs branch of the enterprise-specific MIB {enterprise 2636} and has an object identifier of {jnxMIB 16}. The SCU MIB includes one object, jnxScuStats, which has an object identifier of {jnxScu 1}. For information about configuring source and destination class usage, see the *JUNOS Internet Software Configuration Guide: Policy Framework* and *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-scu.txt.

This chapter discusses the following topic:

jnxScuStatsTable

The entries in the jnxScuStatsTable, whose object identifier is {jnxScuStats 1}, are represented by jnxScuStatsEntry, whose object identifier is {jnxScuStatsTable 1}, and are listed in the following table:

Table 49: jnxScuStatsEntry

Object	Object Identifier	Description
jnxScuStatsDstIfIndex	jnxScuStatsEntry 1	The destination interface index, which is the egress interface of traffic that is counted by this table entry.
jnxScuStatsAddrFamily	jnxScuStatsEntry 2	The address family of an entry's traffic in IPv4 format.
jnxScuStatsClassName	jnxScuStatsEntry 3	The name of the source class. All traffic counted in this table entry satisfies the requirements defined by this source class.

Object	Object Identifier	Description
jnxScuStatsPackets	jnxScuStatsEntry 4	The number of packets sent out of jnxScuStatsDstIfIndex that match the source class (jnxScuStatsClassName) and the address type (jnxScuStatsAddrFamily) defined for a table entry.
jnxScuStatsBytes	jnxScuStatsEntry 5	The number of bytes sent out of jnxScuStatsDstIfIndex that match the source class (jnxScuStatsClassName) and the address type (jnxScuStatsAddrFamily) defined for a table entry.
jnxScuStatsClName	jnxScuStatsEntry 6	The name of the source class. This object is a duplicate of jnxScuStatsClassName and is included to satisfy those network management applications that cannot extract the class name from the instance portion of the OID.

Chapter 20
Interpret the Passive Monitoring MIB

The passive monitoring MIB, whose object identifier is {jnxMibs 19}, performs traffic flow monitoring and lawful interception of packets transiting between two routers. This MIB allows you to do the following:

- Gather and export detailed information about IPv4 traffic flows between source and destination nodes in your network.
- Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.
- The passive monitoring MIB has three tables: jnxPMonFlowTable, JnxPMonErrorTable, and jnxPMonMemoryTable. jnxPMonFlowTable monitors and collects statistics on the flow of traffic on a Passive Monitoring PIC. jnxPMonErrorTable monitors and collects statistics on packet and memory errors on a Passive Monitoring PIC. jnxPMonMemoryTable monitors and collects statistics on memory usage on a Passive Monitoring PIC. For information about system requirements, see the *JUNOS Internet Software Feature Guide*. To view this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-pmon.txt.

This chapter documents only jnxPMonFlowTable.

This chapter contains the following topic:

jnxPMonFlowTable

jnxPMonFlowTable has an object identifier of {jnxPMon 1}. Its entries are represented by JnxPMonFlowEntry, which contains the following objects:

Table 50: jnxPMonFlowEntry

Object	Object Identifer	Description
jnxPMonCurrentActiveFlows	jnxPMonFlowEntry 1	Monitors the number of currently active flows on a Passive Monitoring PIC.
jnxPMonTotalFlows	jnxPMonFlowEntry 2	Monitors the total flows on a Passive Monitoring PIC.
jnxPMonTotalFlowsPackets	jnxPMonFlowEntry 3	Monitors the total packet flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowPackets	jnxPMonFlowEntry 4	Monitors the number of packets in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsBytes	jnxPMonFlowEntry 5	Monitors the number of total of bytes in all flows on a Passive Monitoring PIC.
jnxPMonTenSecondAverageFlowBytes	jnxPMonFlowEntry 6	Monitors the number of bytes in all flows in a 10-second average on a Passive Monitoring PIC.
jnxPMonTotalFlowsExpired	jnxPMonFlowEntry 7	Monitors the number of total flows expired on a Passive Monitoring PIC.
jnxPMonTotalFlowsAged	jnxPMonFlowEntry 8	Monitors the number of total flows aged on a Passive Monitoring PIC.
jnxPMonTotalFlowsExported	jnxPMonFlowEntry 9	Monitors the number of total flows exported on a Passive Monitoring PIC.
jnxPMonTotalFlowsPacketsExported	jnxPMonFlowEntry 10	Monitors the number of total flow packets exported on a Passive Monitoring PIC.

Chapter 21

Interpret the SONET/SDH Interface Management MIB

The SONET/SDH Interface Management MIB sends the current alarm state for each SONET/SDH interface. When the alarm state changes on an interface, the MIB updates its alarm status. To view this MIB, see www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/mib-jnx-sonet.txt.

This chapter discusses the following topic:

jnxSonetAlarmsTable

The jnxSonetAlarmsTable, whose object identifier is {jnxSonetAlarm 1}, provides information about alarm status on SONET/SDH physical interfaces.

jnxSonetAlarmEntry

The jnxSonetAlarmEntry, whose object identifier is {jnxSonetAlarmTable 1} has five objects, which are listed in table 51:

Table 51: jnxSonetAlarmTable

Object	Object Identifier	Description
jnxSonetCurrentAlarms	JnxSonetAlarmEntry 1	Identifies all the active SONET/SDH alarms on this interface.
jnxSonetLastAlarmId	JnxSonetAlarmEntry 2	Identifies the SONET/SDH alarm that most recently was set or cleared.
jnxSonetLastAlarmTime	JnxSonetAlarmEntry 3	The value of sysUpTime when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmDate	JnxSonetAlarmEntry 4	The system date and time when the management subsystem learned of the last alarm event.
jnxSonetLastAlarmEvent	JnxSonetAlarmEntry 5	Indicates whether the last alarm event set a new alarm or cleared an existing alarm.

Table 52 provides an example of jnxSonetAlarmInterface objects on an M20 router.

Table 52: jnxSonetAlarmInterface Objects in jnxSonetAlarmTable of an M20 router

Alarm Interface	CurrentAlarms	Last Alarm ID	Last Alarm Time (System Up Time)	Last Alarm Date and Time	Last Alarm Event
14	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.15	2002-10-15, 10:21:14.0,-7:0	set(2)
15	sonetLosAlarm(3)	sonetLosAlarm(3)	8 days, 4:09:46.22	2002-10-23,14:29:23.0,-7:0	set(2)
16	sonetLolAlarm(0) sonetLosAlarm(3)	sonetBerrSdAlarm(8)	8 days, 4:09:46.21	2002-10-23,14:29:23.0,-7:0	cleared(3)
17	sonetLofAlarm(2)	sonetLaisAlarm(5)	8 days, 4:09:47.21	2002-10-23,14:29:24.0,-7:0	cleared(3)
18	----------	sonetLosAlarm(3)	7 days, 4:31:27.53	2002-10-22,14:51:4.0,-7:0	cleared(3)
19	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.16	2002-10-15,10:21:14.0,-7:0	set(2)
20	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.17	2002-10-15,10:21:14.0,-7:0	set(2)
21	----------	sonetLofAlarm(2)	7 days, 11:15:00.15	2002-10-22,21:34:37.0,-7:0	cleared(3)
22	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLolAlarm(0)	7 days, 6:33:32.02	2002-10-22,16:53:8.0,-7:0	set(2)
23	----------	sonetLosAlarm(3)	7 days, 6:33:45.02	2002-10-22,16:53:21.0,-7:0	cleared(3)
24	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.07	2002-10-15,10:21:14.0,-7:0	set(2)
25	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:37.08	2002-10-15,10:21:14.0,-7:0	set(2)
26	----------	----------	0:00:00.00	0-0-0,0:0:0.0,	none(1)
27	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
28	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)
29	sonetLolAlarm(0) sonetLosAlarm(3)	sonetLosAlarm(3)	0:01:38.04	2002-10-15,10:21:14.0,-7:0	set(2)

Part 5

Accounting Options

Chapter 22
Accounting Options Overview

An accounting profile represents common characteristics of collected accounting data, including the following:

- Collection interval
- File to contain accounting data
- Specific fields and counter names on which to collect statistics

You can configure multiple accounting profiles as described in Table 53.

Table 53: Types of Accounting Profiles

Type of Profile	Description
Interface profile	Collects the specified error and statistic information.
Filter profile	Collects the byte and packet counts for the counter names specified in the filter profile.
Routing Engine profile	Collects selected Routing Engine statistics and logs them to a specified file.
Class usage profile	Collects class usage statistics and logs them to a specified file.

Chapter 23
Configure Accounting Options

To configure accounting options, include statements at the [edit accounting-options] hierarchy level of the configuration.

```
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        destination-classes {
            destination-class-name;
        }
        source-classes {
            source-class-name;
    }
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
    }
    filter-profile profile-name {
        counters {
            counter-name;
        }
        file filename;
        interval minutes;
        }
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
    routing-engine-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
}
```

By default, accounting options are disabled.

This section describes the minimum required configuration and discusses the following tasks for configuring accounting options:

Minimum Accounting Options Configuration

To enable accounting options on the router, you must perform at least the following tasks:

- Configure accounting options by including a file statement and one or more source-class-usage, destination-class-profile, filter-profile, interface-profile, or routing-engine-profile statements at the [edit accounting-options] hierarchy level:

```
[edit]
accounting-options {
    class-usage-profile profile-name {
        file filename;
        interval minutes;
        source-classes {
            source-class-name;
        destination-classes {
            destination-class-name;
        }
    }
    file filename {
        archive-sites {
            site-name;
        }
        files filenumber;
        size bytes;
        transfer-interval minutes;
    }
    filter-profile profile-name {
        counters {
            counter-name;
        }
        file filename;
        interval minutes;
    }
    interface-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
```

```
    routing-engine-profile profile-name {
        fields {
            field-name;
        }
        file filename;
        interval minutes;
    }
}
```

- Apply the profiles to the chosen interfaces or filters.

  Apply an interface profile to a physical or logical interface by including the accounting-profile statement at either the [edit interfaces *interface-name*] or the [edit interfaces *interface-name* unit *number*] hierarchy level. For more information on interface profiles, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service.*

```
[edit interfaces]
interface-name {
    accounting-profile profile-name;
    unit number {
        accounting-profile profile-name;
    }
}
```

You do not apply destination class profiles to interfaces. Although the interface needs to have the destination-class-usage statement configured, the destination class profile automatically finds all interfaces with the destination class configured.

  Apply a filter profile to a firewall filter by including the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level:

```
[edit firewall]
filter filter-name {
    accounting-profile profile-name;
}
```

  You do not need to apply the Routing Engine profile to an interface because the statistics are collected on the Routing Engine itself.

Configure Files

An accounting profile specifies what statistics should be collected and written to a log file. To configure an accounting-data log file, include the file statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
   file filename {
      archive-sites {
         site-name;
      }
      file filenumber;
      size bytes;
      transfer-interval minutes;
      }
   }
```

If the filename contains spaces, enclose it in quotation marks (" "). The filename cannot contain a forward slash (/). The file is created in the /var/log directory and can contain data from multiple profiles.

All accounting-data log files include header and trailer sections that start with a # in the first column. The header contains the file creation time, the hostname, and the columns that appear in the file. The trailer contains the time that the file was closed.

Whenever any configured value changes that affects the columns in a file, the file creates a new profile layout record that contains a new list of columns.

You must configure the file size; all other properties are optional.

- Configure the Maximum Size of the File on page 236
- Configure the Maximum Number of Files on page 237
- Configure the Transfer Interval of the File on page 237
- Configure Archive Sites on page 237

Configure the Maximum Size of the File

To configure the maximum size of the files, include the size statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
size bytes;
```

The size statement is the maximum size of the log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). The minimum value for *bytes* is 256 KB. You must configure *bytes*; the remaining attributes are optional.

Configure the Maximum Number of Files

To configure the maximum number of the files, include the file statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
files filenumber;
```

The files statement specifies the maximum number of files. When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. The minimum value for *filenumber* is 3 and the default value is 10.

Configure the Transfer Interval of the File

To configure the transfer interval of the files, include the transfer-interval statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
transfer-interval minutes;
```

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Configure Archive Sites

After a file reaches its maximum size or the transfer-interval time is exceeded, the file is closed, renamed, and, if you configured an archive site, transferred to a remote host. To configure archive sites, include the archive-sites statement at the [edit accounting-options file *filename*] hierarchy level:

```
[edit accounting-options file filename]
archive-sites {
    site-name;
}
```

site-name is any valid FTP URL. For more information on how to specify valid FTP URLs, see the *JUNOS Internet Software Configuration Guide: Getting Started*. You can specify more than one URL, in any order. When a file is archived, the router attempts to transfer the file to the first URL in the list, trying the next site in the list only if the transfer does not succeed. The log file is stored at the archive site with a filename of the format *router-name_log-filename_timestamp*.

Configure the Interface Profile

An interface profile specifies the information collected and written to a log file. You can configure a profile to collect error and statistic information for input and output packets on a particular physical or logical interface.

To configure an interface profile, include the interface-profile statement at the [edit accounting-options] hierarchy:

```
[edit accounting-options]
interface-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

Each accounting profile must have a unique *profile-name*. To apply a profile to a physical or logical interface, include the accounting-profile statement at either the [edit interfaces *interface-name*] or the [edit interfaces *interface-name* unit *number*] hierarchy level. For more information, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service*.

To configure an interface profile, you perform the tasks described in the following sections:

Configure Fields

An interface profile must specify what statistics are collected. To configure which statistics should be collected for an interface, include the fields statement at the [edit accounting options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
fields {
    field-name;
}
```

Configure the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
file filename;
```

You must specify a filename statement for the interface profile that has already been configured at the [edit accounting-options] hierarchy level.

Configure the Interval

Each interface with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options interface-profile *profile-name*] hierarchy level:

```
[edit accounting-options interface-profile profile-name]
interval minutes;
```

The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of interfaces might cause serious performance degradation.

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configure the Interface Profile

Configure the interface profile:

```
[edit]
accounting-options {
    file if_stats {
        size 40 files 5;
    }
    interface-profile if_profile1 {
        file if_stats;
        interval 30;
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-multicast;
            output-multicast;
        }
    }
    interface-profile if_profile2 {
        file if_stats;
        interval 30;
        fields {
            input-bytes;
            output-bytes;
            input-packets;
            output-packets;
            input-multicast;
            output-multicast;
        }
    }
interfaces {
    ge-1/0/0 {
        accounting-profile if_profile1;
        unit 0 {
            accounting-profile if_profile2;
            …
```

```
        }
      }
    }
```

The two interface profiles, if-profile1 and if-profile2, write data to the same file, if-stats. The if-stats file might look like the following:

```
#FILE CREATED 976823478 2000-12-14-19:51:18
#hostname host
#profile-layout
if_profile2,epoch-timestamp,interface-name,snmp-index,input-bytes,output-bytes,
input-packets,output-packets,input-multicast,output-multicast
#profile-layout
if_profile1,epoch-timestamp,interface-name,snmp-index,input-bytes,output-bytes,
input-packets
if_profile2,976823538,ge-1/0/0.0,8,134696815,3681534,501088,40723,0,0
if_profile1,976823538,ge-1/0/0,7,134696815,3681534,501088
...
#FILE CLOSED 976824378 2000-12-14-20:06:18
```

Configure the Filter Profile

A filter profile specifies error and statistics information collected and written to a file. A filter profile must specify for which counter names statistics are collected. To configure a filter profile, include the filter-profile statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
filter-profile profile-name {
    counters {
        counter-name;
    }
    file filename;
    interval minutes;
}
```

To apply the filter profile, include the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level. For more information on firewall filters, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service*.

To configure a filter profile, you can perform the tasks described in the following sections:

- Configure the Counters on page 241
- Configure the File Information on page 241
- Configure the Interval on page 241
- Example: Configure a Filter Profile on page 242
- Example: Configure Interface-Specific Firewall Counters and Filter Profiles on page 243

Configure the Counters

Statistics are collected for all counters specified in the filter profile. To configure the counters, include the counters statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
counters {
    counter-name;
}
```

Configure the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
file filename;
```

You must specify a filename for the filter profile that has already been configured at the [edit accounting options] hierarchy level.

If the configured file size or transfer interval is exceeded, the JUNOS software closes the file and starts a new one. By default, the transfer interval value is 30 minutes. If the transfer interval is not configured, the JUNOS software closes the file and starts a new one when the file size exceeds its configured value or the default transfer interval value exceeds 30 minutes. To avoid transferring files every 30 minutes, specify a different value for the transfer interval.

Configure the Interval

Each filter with an accounting profile enabled has statistics collected once per interval time specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options filter-profile *profile-name*] hierarchy level:

```
[edit accounting-options filter-profile profile-name]
interval minutes;
```

The minimum interval allowed is 1 minute. Configuring a low interval in an accounting profile for a large number of filters might cause serious performance degradation.

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configure a Filter Profile

Configure a filter profile:

```
[edit]
accounting-options {
    file fw_accounting {
        size 500k files 4;
    }
    filter-profile fw_profile1 {
        file fw_accounting;
        interval 60;
        counters {
            counter1;
            counter2;
            counter3;
        }
    }
}
firewall {
    filter myfilter {
        accounting-profile fw_profile1;
        ...
        term accept-all {
        then {
            count counter1;
            accept;
        }
    }
}
```

The filter profile, fw-profile1, writes data to the file fw_accounting. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#hostname host
#profile-layout
fw_profile1,epoch-timestamp,filter-name,counter-name,packet-count,byte-count
fw_profile1,976826058,myfilter,counter1,163,10764
...
#FILE CLOSED 976826178 2000-12-14-20:36:18
```

Example: Configure Interface-Specific Firewall Counters and Filter Profiles

To collect and log count statistics collected by firewall filters on a per-interface basis, you must configure a filter profile and include the interface-specific statement at the [edit firewall filter *filter-name*] hierarchy level:

Configure the firewall filter accounting profile:

```
[edit accounting-options]
   file cust1_accounting {
   size 500k;
}
filter-profile cust1_profile {
   file cust1_accounting;
   interval 1;
   counters {
      r1;
   }
}
```

Configure the interface-specific firewall counter:

```
[edit firewall]
filter f3 {
   accounting-profile cust1_profile;
   interface-specific;
   term f3-term {
      then {
         count r1;
            accept;
      }
   }
}
```

Apply the firewall filter to an interface:

```
[edit interfaces]
ge-1/0/0 {
   unit 0 {
      family inet {
         filter {
            input f3;
            output f3;
         }
         address 20.20.20.30/24;
      }
   }
}
```

The following example shows the contents of the cust1_accounting file in the /var/log folder that might result from the preceding configuration:

```
#FILE CREATED 995495212 2001-07-18-22:26:52
#hostname host
#profile-layout cust1_profile,epoch-timestamp,interfaces,filter-name,
counter-name,packet-count,byte-count
cust1_profile,995495572,ge-1/0/0.0,f3-ge-1/0/0.0-i,r1-ge-1/0/0.0-i,5953,1008257
cust1_profile,995495602,ge-1/0/0.0,f3-ge-1/0/0.0-o,r1-ge-1/0/0.0-o,5929,1006481
...
```

If the interface-specific statement is not included in the configuration, the following output might result:

```
#FILE CREATED 995495212 2001-07-18-22:26:52
#hostname host
#profile-layout cust1_profile,epoch-timestamp,interfaces,filter-name,
counter-name,packet-count,byte-count
cust1_profile,995495572,ge-1/0/0.0,f3,r1,5953,1008257
cust1_profile,995495632,ge-1/0/0.0,f3,r1,5929,1006481
```

Configure Source Class Usage Options

You can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as *source classes* and *destination classes*. You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces.

Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

For more information about source class usage, see the *JUNOS Internet Software Configuration Guide: Policy Framework*, *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*, and *JUNOS Internet Software Feature Guide*.

To configure source class usage options, perform the following tasks described in this section:

- Configure SCU and/or DCU
- Configure SCU on a Virtual Loopback Interface on page 246
- Configure Class Usage Profiles on page 248

Configure SCU and/or DCU

To configure SCU and/or DCU, perform the following tasks described in this section:

- Create Prefix Route Filters in a Policy Statement on page 245
- Apply the Policy to the Forwarding Table on page 245
- Enable Accounting on Inbound and Outbound Interfaces on page 245

Create Prefix Route Filters in a Policy Statement

```
[edit policy-options]
policy-statement scu-1 {
    term term1
        from {
            route-filter 192.168.1.0/24 orlonger;
        }
        then source-class gold;
    }
}
```

Apply the Policy to the Forwarding Table

```
[edit]
routing-options {
    forwarding-table {
            export scu-1;
        }
    }
}
```

Enable Accounting on Inbound and Outbound Interfaces

You can enable accounting inbound and outbound interfaces:

```
[edit]
interfaces {
    so-6/1/0 {
        unit 0 {
            family inet;
                accounting {
                    destination-class-usage;
                    source-class-usage {
                        output;
                    }
                }
            }
        }
    }
}

[edit]
interfaces {
    ge-0/1/0 {
        unit 0 {
            family inet6 {
                accounting {
                    source-class-usage {
                        input;
                    }
                }
            }
        }
    }
}
```

Optionally, you can include the input and output statements on a single interface:

```
[edit]
interfaces {
    ge-0/1/2 {
        unit 0 {
            family inet6 {
                accounting {
                    source-class-usage {
                        input;
                        output;
                    }
                }
            }
        }
    }
}
```

For more information on configuring route filters and source classes in a routing policy, see the *JUNOS Software Configuration Guide: Policy Framework* and the *JUNOS Software Configuration Guide: Interfaces and Class of Service.*

Configure SCU on a Virtual Loopback Interface

To configure source class usage on the virtual loopback interface, perform the tasks described in the following sections:

- Configure a Virtual Loopback Interface on a Provider Edge Router Equipped with a Tunnel PIC on page 246
- Map the VRF Instance Type to the Virtual Loopback Interface on page 247
- Send Traffic Received From the Virtual Loopback Interface Out the Source Class Output Interface on page 248

Configure a Virtual Loopback Interface on a Provider Edge Router Equipped with a Tunnel PIC

```
[edit interfaces]
vt-0/3/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    input;
                }
            }
        }
    }
}
```

Map the VRF Instance Type to the Virtual Loopback Interface

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface at-2/1/1.0;
        interface vt-0/3/0.0;
        route-distinguisher 10.255.14.225:100;
        vrf-import import-policy-name;
        vrf-export export-policy-name;
        protocols {
            bgp {
                group to-r4 {
                    local-address 10.27.253.1;
                    peer-as 400;
                    neighbor 10.27.253.2;
                }
            }
        }
    }
}
```

For SCU and DCU to work, you must not include the vrf-table-label statement at the [edit routing-instances instance-name] hierarchy level.

Send Traffic Received From the Virtual Loopback Interface Out the Source Class Output Interface

```
[edit interfaces]
at-1/1/0 {
    unit 0 {
        family inet {
            accounting {
                source-class-usage {
                    output;
                }
            }
        }
    }
```

For more information on configuring source class usage on the virtual loopback interface, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service*.

Configure Class Usage Profiles

To collect class usage statistics, perform the tasks described in the following sections:

- Configure a Class Usage Profile on page 248
- Configure the File Information on page 249
- Configure the Interval on page 249
- Create a Class Usage Profile to Collect Source Class Usage Statistics on page 249
- Create a Class Usage Profile to Collect Destination Class Usage Statistics on page 250

Configure a Class Usage Profile

You can configure the class usage profile to collect statistics for particular source and destination classes.

To configure the class usage profile to filter by source classes, include the source-classes statement at the [edit accounting options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
source-classes {
    source-class-name;
}
```

To configure the class usage profile to filter by destination classes, include the destination-classes statement at the [edit accounting options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
destination-classes {
    destination-class-name;
}
```

Configure the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting-options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
file filename;
```

You must specify a filename for the source class usage profile that has already been configured at the [edit accounting options] hierarchy level. You can also specify a filename for the destination class usage profile configured at the [edit accounting options] hierarchy level.

Configure the Interval

Each interface with a class usage profile enabled has statistics collected once per interval specified for the accounting profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options class-usage-profile *profile-name*] hierarchy level:

```
[edit accounting-options class-usage-profile profile-name]
interval minutes;
```

Create a Class Usage Profile to Collect Source Class Usage Statistics

To create a class usage profile to collect source class usage statistics:

```
[edit]
accounting-options {
    class-usage-profile scu-profile1;
    file usage-stats;
    interval 15;
    source-classes {
        gold;
        silver;
        bronze
        }
    }
}
```

The class usage profile, scu-profile1, writes data to the file usage_stats. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, scu_profile,epoch-timestamp,interface-name,source-class,
packet-count,byte-count
scu_profile,980313078,ge-1/0/0.0,gold,82,6888
scu_profile,980313078,ge-1/0/0.0,silver,164,13776
scu_profile,980313078,ge-1/0/0.0,bronze,0,0
scu_profile,980313678,ge-1/0/0.0,gold,82,6888
scu_profile,980313678,ge-1/0/0.0,silver,246,20664
scu_profile,980313678,ge-1/0/0.0,bronze,0,0
```

Create a Class Usage Profile to Collect Destination Class Usage Statistics

To create a class usage profile to collect destination class usage statistics:

```
[edit]
accounting-options {
    class-usage-profile dcu-profile1;
    file usage-stats
    interval 15;
    destination-classes {
        gold;
        silver;
        bronze
        }
    }
}
```

The class usage profile, dcu-profile1, writes data to the file usage-stats. The file might look like the following:

```
#FILE CREATED 976825278 2000-12-14-20:21:18
#profile-layout, dcu_profile,epoch-timestamp,interface-name,destination-class,
packet-count,byte-count
dcu_profile,980313078,ge-1/0/0.0,gold,82,6888
dcu_profile,980313078,ge-1/0/0.0,silver,164,13776
dcu_profile,980313078,ge-1/0/0.0,bronze,0,0
dcu_profile,980313678,ge-1/0/0.0,gold,82,6888
dcu_profile,980313678,ge-1/0/0.0,silver,246,20664
dcu_profile,980313678,ge-1/0/0.0,bronze,0,0
...
#FILE CLOSED 976826178 2000-12-14-20:36:18
```

Configure the Routing Engine Profile

The Routing Engine profile collects Routing Engine statistics and logs them to a file. The Routing Engine profile specifies the fields for which statistics are collected.

To configure a Routing Engine profile, include the routing-engine-profile statement at the [edit accounting-options] hierarchy level:

```
[edit accounting-options]
routing-engine-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

To configure a Routing Engine profile, perform the tasks described in the following sections:

- Configure Fields on page 251
- Configure the File Information on page 251
- Configure the Interval on page 251
- Example: Configure a Routing Engine Profile on page 251

Configure Fields

A Routing Engine profile must specify what statistics are collected. To configure which statistics should be collected for the Routing Engine, include the fields statement at the [edit accounting options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
fields {
    field-name;
}
```

Configure the File Information

Each accounting profile logs its statistics to a file in the /var/log directory.

To configure which file to use, include the file statement at the [edit accounting options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
file filename;
```

You must specify a *filename* for the Routing Engine profile that has already been configured at the [edit accounting-options] hierarchy level.

Configure the Interval

A Routing Engine profile has statistics collected once per interval time specified for the profile. Statistics collection time is scheduled evenly over the configured interval. To configure the interval, include the interval statement at the [edit accounting-options routing-engine-profile *profile-name*] hierarchy level:

```
[edit accounting-options routing-engine-profile profile-name]
interval minutes;
```

The range for interval is 1 through 2880 minutes. The default is 30 minutes.

Example: Configure a Routing Engine Profile

Configure a Routing Engine profile:

```
[edit accounting-options]
file my-file {
    size 300k;
}
routing-engine-profile profile-1 {
    file my-file;
    fields {
        host-name;
        date;
        time-of-day;
        uptime;
        cpu-load-1;
        cpu-load-5;
        cpu-load-15;
    }
}
```

Chapter 24

Summary of Accounting Options Configuration Statements

The following sections explain each of the accounting options configuration statements. The statements are organized alphabetically.

accounting-options

Syntax accounting-options {...}

Hierarchy Level [edit]

Description Configure options for accounting statistics collection.

Usage Guidelines See “Configure Accounting Options” on page 233.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

archive-sites

Syntax
```
archive-sites {
    site-name;
}
```

Hierarchy Level [edit accounting-options file *filename*]

Description Configure an archive site. If more than one site name is configured, an ordered list of archive sites for the accounting-data log files is created. When a file is archived, the router attempts to transfer the file to the first URL in the list, moving to the next site only if the transfer does not succeed. The log file is stored at the archive site with a filename of the format *router-name_log-filename_timestamp*.

Options *site-name*—Any valid FTP URL to a destination. For information on how to specify valid FTP URLs, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Usage Guidelines See “Configure Archive Sites” on page 237.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

class-usage-profile

Syntax
```
class-usage-profile profile-name {
    file filename;
    interval minutes;
    source-classes {
        source-class-name;
    }
    destination-classes {
        destination-class-name
    }
}
```

Hierarchy Level [edit accounting-options]

Description Create a class usage profile, which is used to log class usage statistics to a file in the /var/log directory. The class usage profile logs class usage statistics for the configured source classes on every interface that has destination-class-usage configured.

For information on configuring source classes, see the *JUNOS Software Configuration Guide: Routing and Routing Protocols*. For information on configuring source class usage, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service*.

Options *profile-name*—Name of the destination class profile.

The remaining statements are explained separately.

Usage Guidelines See “Configure Class Usage Profiles” on page 248.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

counters

Syntax
```
counters {
    counter-name;
}
```

Hierarchy Level [edit accounting-options filter-profile *profile-name*]

Description Names of counters for which filter profile statistics are collected. The packet and byte counts for the counters are logged to a file in the /var/log directory.

Options *counter-name*—Name of the counter.

Usage Guidelines See “Configure the Counters” on page 241.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

destination-classes

Syntax destination-classes
destination-class-name;

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*]

Description Specify the destination classes for which statistics are collected.

Options *destination-class-name*—Name of the destination class to include in the source class usage profile.

Usage Guidelines See “Configure a Class Usage Profile” on page 248.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fields

fields (for interface profiles)

Syntax fields {
field-name;
}

Hierarchy Level [edit accounting-options interface-profile *profile-name*]

Description Statistics to collect in an accounting-data log file for an interface.

Options *field-name*—Name of the field:

- input-bytes—Input bytes
- input-errors—Generic input error packets
- input-multicast—Input packets arriving by multicast
- input-packets—Input packets
- input-unicast—Input unicast packets
- output-bytes—Output bytes
- output-errors—Generic output error packets
- output-multicast—Output packets sent by multicast
- output-packets—Output packets
- output-unicast—Output unicast packets

Usage Guidelines See “Configure the Interface Profile” on page 238.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

fields (for Routing Engine profiles)

Syntax
```
fields {
    field-name;
}
```

Hierarchy Level [edit accounting-options routing-engine-profile *profile-name*]

Description Statistics to collect in an accounting-data log file for a Routing Engine.

Options *field-name*—Name of the field:

- cpu-load-1—Average system load over the last 1 minute
- cpu-load-5—Average system load over the last 5 minutes
- cpu-load-15—Average system load over the last 15 minutes
- date—Date in YYYYMMDD format
- host-name—Hostname for the router
- time-of-day—Time of day in HHMMSS format
- uptime—Time since last reboot in seconds

Usage Guidelines See “Configure the Routing Engine Profile” on page 250.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

file

file (create a log file)

Syntax

```
file filename {
    archive-sites {
        site-name;
    }
    files number
    size bytes
    transfer-interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Description Information on a log file used for accounting data.

Options files *number*—The maximum number of files. When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. The minimum value for *number* is 3 and the default value is 10.

The remaining statements are explained separately.

Usage Guidelines See "Configure the Maximum Number of Files" on page 237.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

file (for a profile to use)

Syntax

```
file filename;
```

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*],
[edit accounting-options filter-profile *profile-name*],
[edit accounting-options interface-profile *profile-name*],
[edit accounting-options routing-engine-profile *profile-name*]

Description The accounting log file to use.

Options *filename*—Name of the log file. You must specify a *filename* already configured in the file statement at the [edit accounting-options] hierarchy level.

Usage Guidelines See "Configure the Interface Profile" on page 238, "Configure the Filter Profile" on page 240, and "Configure the Routing Engine Profile" on page 250.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

filter-profile

Syntax

```
filter-profile profile-name {
    counters {
        counter-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Description Create a profile to filter and collect packet and byte count statistics and write them to a file in the /var/log directory. To apply the profile to a firewall filter, you include the accounting-profile statement at the [edit firewall filter *filter-name*] hierarchy level. For more information on firewall filters, see the *JUNOS Software Configuration Guide: Interfaces and Class of Service*.

Options *profile-name*—Name of the filter profile.

The remaining statements are explained separately.

Usage Guidelines See “Configure the Filter Profile” on page 240.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

interface-profile

Syntax

```
interface-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Description Create a profile to filter and collect error and packet statistics and write them to a file in the /var/log directory. You can specify an interface profile for either a physical or a logical interface.

Options *profile-name*—Name of the interface profile.

The remaining statements are explained separately.

Usage Guidelines See “Configure the Interface Profile” on page 238.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

interval

Syntax interval *minutes*;

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*],
[edit accounting-options filter-profile *profile-name*],
[edit accounting-options interface-profile *profile-name*],
[edit accounting-options routing-engine-profile *profile-name*]

Description How often statistics are collected for the accounting profile.

Options *minutes*—Amount of time between each collection of statistics.
Range: 1 through 2880 minutes
Default: 30 minutes

Usage Guidelines See "Configure the Interface Profile" on page 238, "Configure the Filter Profile" on page 240, and "Configure the Routing Engine Profile" on page 250.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

routing-engine-profile

Syntax
```
routing-engine-profile profile-name {
    fields {
        field-name;
    }
    file filename;
    interval minutes;
}
```

Hierarchy Level [edit accounting-options]

Description Create a Routing Engine profile to collect selected Routing Engine statistics and write them to a file in the /var/log directory.

Options *profile-name*—Name of the Routing Engine statistics profile.

The remaining statements are explained separately.

Usage Guidelines See "Configure the Routing Engine Profile" on page 250.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

size

Syntax size *bytes*;

Hierarchy Level [edit accounting-options file *filename*]

Description Attributes of an accounting-data log file.

Options size *bytes*—Maximum size of each log file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a log file (for example, profilelog) reaches its maximum size, it is renamed profilelog.0, then profilelog.1, and so on, until the maximum number of log files is reached. Then the oldest log file is overwritten. If you do not specify a size, the file is closed, archived, and renamed when the time specified for the transfer interval is exceeded.
Syntax: *x* to specify bytes, *x*k to specify KB, *x*m to specify MB, *x*g to specify GB
Range: 256 KB through 1 GB

Usage Guidelines See "Configure the Maximum Size of the File" on page 236.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

source-classes

Syntax
```
source-classes {
    source-class-name;
```

Hierarchy Level [edit accounting-options class-usage-profile *profile-name*]

Description Specify the source classes for which statistics are collected.

Options *source-class-name*—Name of the source class to include in the class usage profile.

Usage Guidelines See "Configure a Class Usage Profile" on page 248.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

transfer-interval

Syntax transfer-interval *minutes*;

Hierarchy Level [edit accounting-options file *filename*]

Description Time the file remains open and receiving new statistics before it is closed and transferred to an archive site.

Options transfer-interval *minutes*—Time the file remains open and receiving new statistics before it is closed and transferred to an archive site.
Range: 15 through 2880 minutes
Default: 30 minutes

Usage Guidelines See "Configure the Transfer Interval of the File" on page 237.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Part 6
Appendix

■ Glossary on page 263

Appendix A
Glossary

A

AAL ATM adaptation layer. A series of protocols enabling various types of traffic, including voice, data, image, and video, to run over an ATM network.

active route Route chosen from all routes in the routing table to reach a destination. Active routes are installed into the forwarding table.

add/drop multiplexer *See ADM*.

Address Resolution Protocol *See ARP*.

adjacency Portion of the local routing information that pertains to the reachability of a single neighbor over a single circuit or interface.

ADM Add/drop multiplexer. SONET functionality that allows lower-level signals to be dropped from a high-speed optical connection.

aggregation Combination of groups of routes that have common addresses into a single entry in the routing table.

AH Authentication Header. A component of the IPSec protocol used to verify that the contents of a packet have not been changed, and to validate the identity of the sender. *See also ESP*.

ANSI American National Standards Institute. The United States' representative to the ISO.

APQ Alternate Priority Queuing. Dequeuing method that has a special queue, similar to SPQ, which is visited only 50 percent of the time. The packets in the special queue still have a predictable latency, although the upper limit of the delay is higher than that with SPQ. Since the other configured queues share the remaining 50 percent of the service time, queue starvation is usually avoided. *See also SPQ*.

APS Automatic Protection Switching. Technology used by SONET ADMs to protect against circuit faults between the ADM and a router and to protect against failing routers.

area Routing subdomain that maintains detailed routing information about its own internal composition and that maintains routing information that allows it to reach other routing subdomains. In IS-IS, an area corresponds to a Level 1 subdomain.

In IS-IS and OSPF, a set of contiguous networks and hosts within an autonomous system that have been administratively grouped together.

area border router Router that belongs to more than one area. Used in OSPF.

ARP — Address Resolution Protocol. Protocol for mapping IP addresses to MAC addresses.

AS — Autonomous system. Set of routers under a single technical administration. Each AS normally uses a single interior gateway protocol (IGP) and metrics to propagate routing information within the set of routers. Also called *routing domain*.

AS boundary router — In OSPF, routers that exchange routing information with routers in other ASs.

AS external link advertisements — OSPF link-state advertisement sent by AS boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the AS (except for stub areas).

AS path — In BGP, the route to a destination. The path consists of the AS numbers of all routers a packet must go through to reach a destination.

ASIC — Application-specific integrated circuit. Specialized processors that perform specific functions on the router.

ATM — Asynchronous Transfer Mode. A high-speed multiplexing and switching method utilizing fixed-length cells of 53 octets to support multiple types of traffic.

atomic — Smallest possible operation. An atomic operation is performed either entirely or not at all. For example, if machine failure prevents a transaction from completing, the system is rolled back to the start of the transaction, with no changes taking place.

Authentication Header — *See AH*.

Automatic Protection Switching — *See APS*.

autonomous system — *See AS*.

autonomous system boundary router — In OSPF, routers that exchange routing information with routers in other ASs.

autonomous system external link advertisements — OSPF link-state advertisement sent by autonomous system boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the autonomous system (except for stub areas).

autonomous system path — In BGP, the route to a destination. The path consists of the autonomous system numbers of all the routers a packet must pass through to reach a destination.

B

backbone area — In OSPF, an area that consists of all networks in area ID 0.0.0.0, their attached routers, and all area border routers.

backplane — On an M40 router, component of the Packet Forwarding Engine that distributes power, provides signal connectivity, manages shared memory on FPCs, and passes outgoing data cells to FPCs.

bandwidth — The range of transmission frequencies a network can use, expressed as the difference between the highest and lowest frequencies of a transmission channel. In computer networks, greater bandwidth indicates faster data-transfer rate capacity.

Bellcore — Bell Communications Research. Research and development organization created after the divestiture of the Bell System. It is supported by the regional Bell holding companies (RBHCs), which own the regional Bell operating companies (RBOCs).

BERT — Bit error rate test. A test that can be run on a T3 interface to determine whether it is operating properly.

BGP — Border Gateway Protocol. Exterior gateway protocol used to exchange routing information among routers in different autonomous systems.

bit error rate test — *See BERT*.

BITS — Building Integrated Timing Source. Dedicated timing source that synchronizes all equipment in a particular building.

Border Gateway Protocol — *See BGP*.

broadcast — Operation of sending network traffic from one network node to all other network nodes.

bundle — Collection of software that makes up a JUNOS software release.

C

CB — Control Board. Part of the host subsystem that provides control and monitoring functions for router components.

CCC — Circuit cross-connect. A JUNOS software feature that allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay DLCI, an ATM VC, a PPP interface, a Cisco HDLC interface, or an MPLS label-switched path (LSP).

CE device — Customer edge device. Router or switch in the customer's network that is connected to a service provider's provider edge (PE) router and participates in a Layer 3 VPN.

CFM — Cubic feet per minute. Measure of air flow in volume per minute.

Challenge Handshake Authentication Protocol — *See CHAP*.

channel service unit — *See CSU/DSU*.

CHAP — A protocol that authenticates remote users. CHAP is a server-driven, three-step authentication mechanism that depends on a shared secret password that resides on both the server and the client.

CIDR — Classless interdomain routing. A method of specifying Internet addresses in which you explicitly specify the bits of the address to represent the network address instead of determining this information from the first octet of the address.

CIP — Connector Interface Panel. On an M160 router, the panel that contains connectors for the Routing Engines, BITS interfaces, and alarm relay contacts.

circuit cross-connect — *See CCC*.

class of service — *See CoS*.

CLEC — (Pronounced “see-lek”) Competitive Local Exchange Carrier. Company that competes with the already established local telecommunications business by providing its own network and switching.

CLEI — Common language equipment identifier. Inventory code used to identify and track telecommunications equipment.

CLI Command-line interface. Interface provided for configuring and monitoring the routing protocol software.

client peer In a BGP route reflection, a member of a cluster that is not the route reflector. *See also nonclient peer*.

CLNP Connectionless Network Protocol. ISO-developed protocol for OSI connectionless network service. CLNP is the OSI equivalent of IP.

cluster In BGP, a set of routers that have been grouped together. A cluster consists of one system that acts as a route reflector, along with any number of client peers. The client peers receive their route information only from the route reflector system. Routers in a cluster do not need to be fully meshed.

community In BGP, a group of destinations that share a common property. Community information is included as one of the path attributes in BGP update messages.

confederation In BGP, a group of systems that appears to external autonomous systems to be a single autonomous system.

constrained path In traffic engineering, a path determined using RSVP signaling and constrained using CSPF. The ERO carried in the packets contains the constrained path information.

core The central backbone of the network.

CoS Class of service. The method of classifying traffic on a packet-by-packet basis using information in the ToS byte to provide different service levels to different traffic.

CPE Customer premises equipment. Telephone or other service provider equipment located at a customer site.

craft interface Mechanisms used by a Communication Workers of America craftsperson to operate, administer, and maintain equipment or provision data communications. On a Juniper Networks router, the craft interface allows you to view status and troubleshooting information and perform system control functions.

CSCP Class Selector Codepoint.

CSNP Complete sequence number PDU. Packet that contains a complete list of all the LSPs in the IS-IS database.

CSPF Constrained Shortest Path First. An MPLS algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network.

CSU/DSU Channel service unit/data service unit. Channel service unit connects a digital phone line to a multiplexer or other digital signal device. Data service unit connects a DTE to a digital phone line.

customer edge device *See CE device*.

D

daemon Background process that performs operations on behalf of the system software and hardware. Daemons normally start when the system software is booted, and they run as long as the software is running. In the JUNOS software, daemons are also referred to as processes.

damping Method of reducing the number of update messages sent between BGP peers, thereby reducing the load on these peers without adversely affecting the route convergence time for stable routes.

data circuit-terminating equipment *See DCE*.

data-link connection identifier *See DLCI*.

data service unit *See CSU/DSU*.

Data Terminal Equipment *See DTE*.

dcd The JUNOS software interface process (daemon).

DCE Data circuit-terminating equipment. RS-232-C device, typically used for a modem or printer, or a network access and packet switching node.

default address Router address that is used as the source address on unnumbered interfaces.

denial of service *See DoS*.

dense wavelength-division multiplexing *See DWDM*.

designated router In OSPF, a router selected by other routers that is responsible for sending link-state advertisements that describe the network, which reduces the amount of network traffic and the size of the routers' topological databases.

destination prefix length Number of bits of the network address used for host portion of a CIDR IP address.

DHCP Dynamic Host Configuration Protocol. Allocates IP addresses dynamically so that they can be reused when they are no longer needed.

Diffie-Hellman A public key scheme, invented by Whitfield Diffie and Martin Hellman, used for sharing a secret key without communicating secret information, thus precluding the need for a secure channel. Once correspondents have computed the secret shared key, they can use it to encrypt communications.

Diffserv Differentiated Service (based on RFC 2474). Diffserv uses the ToS byte to identify different packet flows on a packet-by-packet basis. Diffserv adds a Class Selector Codepoint (CSCP) and a Differentiated Services Codepoint (DSCP).

Dijkstra algorithm *See SPF*.

DIMM Dual inline memory module. 168-pin memory module that supports 64-bit data transfer.

direct routes — *See interface routes*.

DLCI — Data-link connection identifier. Identifier for a Frame Relay virtual connection (also called a logical interface).

DoS — Denial of service. System security breach in which network services become unavailable to users.

DRAM — Dynamic random-access memory. Storage source on the router that can be accessed quickly by a process.

drop profile — Drop probabilities for different levels of buffer fullness that are used by RED to determine from which queue to drop packets.

DSCP — Differentiated Services Codepoint.

DSU — Data service unit. A device used to connect a DTE to a digital phone line. Converts digital data from a router to voltages and encoding required by the phone line. *See also CSU/DSU*.

DTE — Data Terminal Equipment. RS-232-C interface that a computer uses to exchange information with a serial device.

DVMRP — Distance Vector Multicast Routing Protocol. Distributed multicast routing protocol that dynamically generates IP multicast delivery trees using a technique called reverse path multicasting (RPM) to forward multicast traffic to downstream interfaces.

DWDM — Dense wavelength-division multiplexing. Technology that enables data from different sources to be carried together on an optical fiber, with each signal carried on its own separate wavelength.

Dynamic Host Configuration Protocol — *See DHCP*.

E

EBGP — External BGP. BGP configuration in which sessions are established between routers in different ASs.

ECSA — Exchange Carriers Standards Association. A standards organization created after the divestiture of the Bell System to represent the interests of interexchange carriers.

edge router — In MPLS, a router located at the beginning or end of a label-switching tunnel. When at the beginning of a tunnel, an edge router applies labels to new packets entering the tunnel. When at the end of a tunnel, the edge router removes labels from packets exiting the tunnel. *See also MPLS*.

EGP — Exterior gateway protocol, such as BGP.

egress router — In MPLS, last router in a label-switched path (LSP). *See also ingress router*.

EIA — Electronic Industries Association. A United States trade group that represents manufacturers of electronics devices and sets standards and specifications.

EMI — Electromagnetic interference. Any electromagnetic disturbance that interrupts, obstructs, or otherwise degrades or limits the effective performance of electronics or electrical equipment.

encapsulating security payload — *See ESP*.

end system In IS-IS, network entity that sends and receives packets.

ERO Explicit Route Object. Extension to RSVP that allows an RSVP PATH message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing.

ESP Encapsulating security payload. A fundamental component of IPSec-compliant VPNs, ESP specifies an IP packet's encryption, data integrity checks, and sender authentication, which are added as a header to the IP packet. *See also AH*.

explicit path *See signaled path*.

Explicit Route Object *See ERO*.

export To place routes from the routing table into a routing protocol.

external BGP *See EBGP*.

external metric A cost included in a route when OSPF exports route information from external autonomous systems. There are two types of external metrics: Type 1 and Type 2. Type 1 external metrics are equivalent to the link-state metric; that is, the cost of the route, used in the internal autonomous system. Type 2 external metrics are greater than the cost of any path internal to the autonomous system.

F

fast reroute Mechanism for automatically rerouting traffic on an LSP if a node or link in an LSP fails, thus reducing the loss of packets traveling over the LSP.

FEAC Far-end alarm and control. T3 signal used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal.

FEB Forwarding Engine Board. In M5 and M10 routers, provides route lookup, filtering, and switching to the destination port.

firewall A security gateway positioned between two different networks, usually between a trusted network and the Internet. A firewall ensures that all traffic that crosses it conforms to the organization's security policy. Firewalls track and control communications, deciding whether to pass, reject, discard, encrypt, or log them. Firewalls also can be used to secure sensitive portions of a local network.

FIFO First in, first out.

flap damping *See damping*.

flapping *See route flapping*.

Flexible PIC Concentrator *See FPC*.

Forwarding Engine Board *See FEB*.

forwarding information base *See forwarding table*.

forwarding table JUNOS software forwarding information base (FIB). The JUNOS routing protocol process installs active routes from its routing tables into the Routing Engine forwarding table. The kernel copies this forwarding table into the Packet Forwarding Engine, which is responsible for determining which interface transmits the packets.

FPC Flexible PIC Concentrator. An interface concentrator on which PICs are mounted. An FPC inserts into a slot in a Juniper Networks router. *See also PIC.*

FRU Field-replaceable unit. Router component that customers can replace onsite.

G

group A collection of related BGP peers.

H

hash A one-way function that takes a message of any length and produces a fixed-length digest. In security, a message digest is used to validate that the contents of a message have not been altered in transit. The Secure Hash Algorithm (SHA-1) and Message Digest 5 (MD5) are commonly used hashes.

Hashed Message Authentication Code *See HMAC*.

HDLC High-level data link control. An International Telecommunication Union (ITU) standard for a bit-oriented data link layer protocol on which most other bit-oriented protocols are based.

HMAC Hashed Message Authentication Code. A mechanism for message authentication that uses cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function—for example, MD5 or SHA-1—in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

hold time Maximum number of seconds allowed to elapse between the time a BGP system receives successive keepalive or update messages from a peer.

host module On an M160 router, provides routing and system management functions of the router. Consists of the Routing Engine and Miscellaneous Control Subsystem (MCS).

host subsystem Provides routing and system-management functions of the router. Consists of a Routing Engine and an adjacent Control Board (CB).

I

IANA Internet Assigned Numbers Authority. Regulatory group that maintains all assigned and registered Internet numbers, such as IP and multicast addresses. *See also NIC*.

IBGP Internal BGP. BGP configuration in which sessions are established between routers in the same ASs.

ICMP Internet Control Message Protocol. Used in router discovery, ICMP allows router advertisements that enable a host to discover addresses of operating routers on the subnet.

IDE Integrated Drive Electronics. Type of hard disk on the Routing Engine.

IEC International Electrotechnical Commission. *See ISO*.

IEEE Institute of Electronic and Electrical Engineers. International professional society for electrical engineers.

IETF	Internet Engineering Task Force. International community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.
IGMP	Internet Group Membership Protocol. Used with multicast protocols to determine whether group members are present.
IGP	Interior gateway protocol, such as IS-IS, OSPF, and RIP.
IKE	Internet Key Exchange. The key management protocol used in IPSec, IKE combines the ISAKMP and Oakley protocols to create encryption keys and security associations.
import	To install routes from the routing protocols into a routing table.
ingress router	In MPLS, first router in a label-switched path (LSP). *See also egress router*.
inter-AS routing	Routing of packets among different ASs. *See also EBGP*.
intercluster reflection	In a BGP route reflection, the redistribution of routing information by a route reflector system to all nonclient peers (BGP peers not in the cluster). *See also route reflection*.
interface routes	Routes that are in the routing table because an interface has been configured with an IP address. Also called *direct routes*.
intermediate system	In IS-IS, network entity that sends and receives packets and that can also route packets.
internal BGP	*See IBGP*.
Internet Key Exchange	*See IKE*.
Internet Protocol Security	*See IPSec*.
Internet Security Association and Key Management Protocol	*See ISAKMP*.
intra-AS routing	The routing of packets within a single AS. *See also IBGP.*
IP	Internet Protocol. The protocol used for sending data from one point to another on the Internet.
IPSec	Internet Protocol Security. The industry standard for establishing VPNs, IPSec comprises a group of protocols and algorithms that provide authentication and encryption of data across IP-based networks.
ISAKMP	Internet Security Association and Key Management Protocol. A protocol that allows the receiver of a message to obtain a public key and use digital certificates to authenticate the sender's identity. ISAKMP is designed to be key exchange independent; that is, it supports many different key exchanges. *See also IKE and Oakley*.
IS-IS	Intermediate System-to-Intermediate System protocol. Link-state, interior gateway routing protocol for IP networks that also uses the shortest-path first (SPF) algorithm to determine routes.

ISO — International Organization for Standardization. Worldwide federation of standards bodies that promotes international standardization and publishes international agreements as International Standards.

ISP — Internet service provider. Company that provides access to the Internet and related services.

ITU — International Telecommunications Union (formerly known as the CCITT). Group supported by the United Nations that makes recommendations and coordinates the development of telecommunications standards for the entire world.

J

jitter — Small random variation introduced into the value of a timer to prevent multiple timer expirations from becoming synchronized.

K

kernel forwarding table — *See forwarding table*.

L

label — In MPLS, 20-bit unsigned integer in the range 0 through 1048575, used to identify a packet traveling along an LSP.

label-switched path (LSP) — Sequence of routers that cooperatively perform MPLS operations for a packet stream. The first router in an LSP is called the *ingress router*, and the last router in the path is called the *egress router*. An LSP is a point-to-point, half-duplex connection from the ingress router to the egress router. (The ingress and egress routers cannot be the same router.)

label switching — *See MPLS*.

label-switching router — *See LSR*.

link — Communication path between two neighbors. A link is *up* when communication is possible between the two end points.

link-state PDU (LSP) — Packets that contain information about the state of adjacencies to neighboring systems.

local preference — Optional BGP path attribute carried in internal BGP update packets that indicates the degree of preference for an external route.

loose — In the context of traffic engineering, a path that can use any route or any number of other intermediate (transit) points to reach the next address in the path. (Definition from RFC 791, modified to fit LSPs.)

LSP — *See label-switched path (LSP)*) *or link-state PDU (LSP)*.

LSR — Label-switching router. A router on which MPLS and RSVP are enabled and is thus capable of processing label-switched packets.

M

martian address — Network address about which all information is ignored.

mask — *See subnet mask*.

MBGP — Multiprotocol BGP. An extension to BGP that allows you to connect multicast topologies within and between BGP ASs.

MBone Internet multicast backbone. An interconnected set of subnetworks and routers that support the delivery of IP multicast traffic. The MBone is a virtual network that is layered on top of sections of the physical Internet.

MCS Miscellaneous Control Subsystem. On an M160 router, provides control and monitoring functions for router components and SONET clocking for the router.

MD5 Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. It is used in AH and ESP. *See also SHA-1*.

MDRR Modified Deficit Round Robin. A method for selecting queues to be serviced.

MED Multiple exit discriminator. Optional BGP path attribute consisting of a metric value that is used to determine the exit point to a destination when all other factors in determining the exit point are equal.

mesh Network topology in which devices are organized in a manageable, segmented manner with many, often redundant, interconnections between network nodes.

Message Digest 5 *See MD5*.

MIB Management Information Base. Definition of an object that can be managed by SNMP.

midplane Forms the rear of the PIC cage on M5 and M10 routers and the FPC card cage on M20 and M160 routers. Provides data transfer, power distribution, and signal connectivity.

Miscellaneous Control Subsystem *See MCS*.

MPLS Multiprotocol Label Switching. Mechanism for engineering network traffic patterns that functions by assigning to network packets short labels that describe how to forward them through the network. Also called *label switching*. *See also traffic engineering*.

MTBF Mean time between failure. Measure of hardware component reliability.

MTU Maximum transfer unit. Limit on segment size for a network.

multicast Operation of sending network traffic from one network node to multiple network nodes.

multicast distribution tree The data path between the sender (host) and the multicast group member (receiver or listener).

multiprotocol BGP *See MBGP*.

Multiprotocol Label Switching *See MPLS*.

N

neighbor Adjacent system reachable by traversing a single subnetwork. An immediately adjacent router. Also called a *peer*.

NET Network entity title. Network address defined by the ISO network architecture and used in CLNS-based networks.

network layer reachability information *See NLRI*.

network link advertisement — An OSPF link-state advertisement flooded throughout a single area by designated routers to describe all routers attached to the network.

Network Time Protocol — *See NTP*.

NIC — Network Information Center. Internet authority responsible for assigning Internet-related numbers, such as IP addresses and autonomous system numbers. *See also IANA*.

NLRI — Network layer reachability information. Information that is carried in BGP packets and is used by MBGP.

nonclient peer — In a BGP route reflection, a BGP peer that is not a member of a cluster. *See also client peer*.

not-so-stubby area — *See NSSA*.

NSAP — Network service access point. Connection to a network that is identified by a network address.

n-selector — Last byte of an nonclient peer address.

NSSA — Not-so-stubby area. In OSPF, a type of stub area in which external routes can be flooded.

NTP — Network Time Protocol. Protocol used to synchronize computer clock times on a network.

O

Oakley — A key determination protocol based on the Diffie-Hellman algorithm that provides added security, including authentication. Oakley was the key-exchange algorithm mandated for use with the initial version of ISAKMP, although various algorithms can be used. Oakley describes a series of key exchanges called “modes” and details the services provided by each; for example, Perfect Forward Secrecy for keys, identity protection, and authentication. *See also ISAKMP*.

OC — Optical Carrier. In SONET, Optical Carrier levels indicate the transmission rate of digital signals on optical fiber.

OSI — Open System Interconnection. Standard reference model for how messages are transmitted between two points on a network.

OSPF — Open Shortest Path First. A link-state IGP that makes routing decisions based on the shortest-path-first (SPF) algorithm (also referred to as the *Dijkstra algorithm*).

P

package — A collection of files that make up a JUNOS software component.

Packet Forwarding Engine — The architectural portion of the router that processes packets by forwarding them between input and output interfaces.

path attribute — Information about a BGP route, such as the route origin, AS path, and next-hop router.

PCI — Peripheral Component Interconnect. Standard, high-speed bus for connecting computer peripherals. Used on the Routing Engine.

PCMCIA — Personal Computer Memory Card International Association. Industry group that promotes standards for credit card-size memory or I/O devices.

PDU — Protocol data unit. IS-IS packets.

PE router — Provider edge router. A router in the service provider's network that is connected to a customer edge (CE) device and that participates in a Virtual Private Network (VPN).

PEC — Policing Equivalence Classes. In traffic policing, a set of packets that is treated the same by the packet classifier.

peer — An immediately adjacent router with which a protocol relationship has been established. Also called a *neighbor*.

Perfect Forward Secrecy — *See PFS*.

PFE — *See Packet Forwarding Engine*.

PFS — A condition derived from an encryption system that changes encryption keys often and ensures that no two sets of keys have any relation to each other. The advantage of PFS is that if one set of keys is compromised, only communications using those keys are at risk. An example of a system that uses PFS is Diffie-Hellman.

Physical Interface Card — *See PIC*.

PIC — Physical Interface Card. A network interface–specific card that can be installed on an FPC in the router.

PIM — Protocol Independent Multicast. A protocol-independent multicast routing protocol. PIM Sparse Mode routes to multicast groups that might span wide-area and interdomain internets. PIM Dense Mode is a flood-and-prune protocol.

PLP — Packet Loss Priority.

PLP bit — Packet Loss Priority bit. Used to identify packets that have experienced congestion or are from a transmission that exceeded a service provider’s customer service license agreement. This bit can be used as part of a router’s congestion control mechanism and can be set by the interface or by a filter.

policing — Applying rate limits on bandwidth and burst size for traffic on a particular interface.

pop — Removal of the last label, by a router, from a packet as it exits an MPLS domain.

PPP — Point-to-Point Protocol. Link-layer protocol that provides multiprotocol encapsulation. It is used for link-layer and network-layer configuration.

precedence bits — The first three bits in the ToS byte. On a Juniper Networks router, these bits are used to sort or classify individual packets as they arrive at an interface. The classification determines the queue to which the packet is directed upon transmission.

preference — Desirability of a route to become the active route. A route with a lower preference value is more likely to become the active route. The preference is an arbitrary value in the range 0 through 255 that the routing protocol process uses to rank routes received from different protocols, interfaces, or remote systems.

preferred address — On an interface, the default local address used for packets sourced by the local router to destinations on the subnet.

primary address — On an interface, the address used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface.

primary interface — Router interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.

Protocol-Independent Multicast — *See PIM*.

provider edge router — *See PE router*.

provider router — Router in the service provider's network that does not attach to a customer edge (CE) device.

PSNP — Partial sequence number PDU. Packet that contains only a partial list of the LSPs in the IS-IS link-state database.

push — Addition of a label or stack of labels, by a router, to a packet as it enters an MPLS domain.

Q

QoS — Quality of service. Performance, such as transmission rates and error rates, of a communications channel or system.

quality of service — *See QoS*.

R

RADIUS — Remote Authentication Dial-In User Service. Authentication method for validating users who attempt to access the router using Telnet.

Random Early Detection — *See RED*.

rate limiting — *See policing*.

RBOC — (Pronounced "are-bock") Regional Bell operating company. Regional telephone companies formed as a result of the divestiture of the Bell System.

RDRAM — RAMBUS dynamic random access memory.

RED — Random Early Detection. Gradual drop profile for a given class that is used for congestion avoidance. RED tries to anticipate incipient congestion and reacts by dropping a small percentage of packets from the head of the queue to ensure that a queue never actually becomes congested.

Rendezvous Point — *See RP*.

Resource Reservation Protocol — *See RSVP*.

RFC — Request for Comments. Internet standard specifications published by the Internet Engineering Task Force.

RFI — Radio frequency interference. Interference from high-frequency electromagnetic waves emanating from electronic devices.

RIP — Routing Information Protocol. Distance-vector interior gateway protocol that makes routing decisions based on hop count.

route flapping — Situation in which BGP systems send an excessive number of update messages to advertise network reachability information.

route identifier — IP address of the router from which a BGP, IGP, or OSPF packet originated.

route reflection — In BGP, configuring a group of routers into a cluster and having one system act as a route reflector, redistributing routes from outside the cluster to all routers in the cluster. Routers in a cluster do not need to be fully meshed.

router link advertisement — OSPF link-state advertisement flooded throughout a single area by all routers to describe the state and cost of the router's links to the area.

routing domain — *See AS*.

Routing Engine — Architectural portion of the router that handles all routing protocol processes, as well as other software processes that control the router's interfaces, some of the chassis components, system management, and user access to the router.

routing instance — A collection of routing tables, interfaces, and routing protocol parameters. The set of interfaces belongs to the routing tables and the routing protocol parameters control the information in the routing tables.

routing table — Common database of routes learned from one or more routing protocols. All routes are maintained by the JUNOS routing protocol process.

RP — For PIM-SM, a core router acting as the root of the distribution tree in a shared tree.

rpd — JUNOS software routing protocol process (daemon). User-level background process responsible for starting, managing, and stopping the routing protocols on a Juniper Networks router.

RPM — Reverse path multicasting. Routing algorithm used by DVMRP to forward multicast traffic.

RSVP — Resource Reservation Protocol. Resource reservation setup protocol designed to interact with integrated services on the Internet.

S

SA — Security association. An IPSec term that describes an agreement between two parties about what rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications.

SAP — Session Announcement Protocol. Used with multicast protocols to handle session conference announcements.

SAR — Segmentation and reassembly. Buffering used with ATM.

SCB — System Control Board. On an M40 router, the part of the Packet Forwarding Engine that performs route lookups, monitors system components, and controls FPC resets.

SCG — SONET Clock Generator. Provides Stratum 3 clock signal for the SONET/SDH interfaces on the router. Also provides external clock inputs.

SDH — Synchronous Digital Hierarchy. CCITT variation of SONET standard.

SDP — Session Description Protocol. Used with multicast protocols to handle session conference announcements.

SDRAM — Synchronous dynamic random access memory.

Secure Hash Algorithm — *See SHA-1*.

secure shell	*See SSH*.
security association	*See SA*.
Security Parameter Index	*See SPI*.
SFM	Switching and Forwarding Module. On an M160 router, a component of the Packet Forwarding Engine that provides route lookup, filtering, and switching to FPCs.
SHA-1	Secure Hash Algorithm. A widely used hash function for use with Digital Signal Standard (DSS). SHA-1 is more secure than MD5.
shortest-path-first algorithm	*See SPF*.
signaled path	In traffic engineering, an explicit path; that is, a path determined using RSVP signaling. The ERO carried in the packets contains the explicit path information.
SIB	Switch Interface Board. Provides the switching function to the destination Packet Forwarding Engine.
simplex interface	An interface that assumes that packets it receives from itself are the result of a software loopback process. The interface does not consider these packets when determining whether the interface is functional.
SNMP	Simple Network Management Protocol. Protocol governing network management and the monitoring of network devices and their functions.
SONET	Synchronous Optical Network. High-speed (up to 2.5 Gbps) synchronous network specification developed by Bellcore and designed to run on optical fiber. STS-1 is the basic building block of SONET. Approved as an international standard in 1988. *See also SDH.*
SPF	Shortest-path first, an algorithm used by IS-IS and OSPF to make routing decisions based on the state of network links. Also called the *Dijkstra algorithm*.
SPI	Security Parameter Index. A portion of the IPSec Authentication Header that communicates which security protocols, such as authentication and encryption, are used for each packet in a VPN connection.
SPQ	Strict Priority Queuing. Dequeuing method that provides a special queue that is serviced until it is empty. The traffic sent to this queue tends to maintain a lower latency and more consistent latency numbers than traffic sent to other queues. *See also APQ.*
SSB	System and Switch Board. On an M20 router, Packet Forwarding Engine component that performs route lookups and component monitoring and monitors FPC operation.
SSH	Secure shell. Software that provides a secured method of logging in to a remote network system.
SSRAM	Synchronous Static Random Access Memory.
static LSP	*See static path*.
static path	In the context of traffic engineering, a static route that requires hop-by-hop manual configuration. No signaling is used to create or maintain the path. Also called a *static LSP.*

STM Synchronous Transport Module. CCITT specification for SONET at 155.52 Mbps.

strict In the context of traffic engineering, a route that must go directly to the next address in the path. (Definition from RFC 791, modified to fit LSPs.)

STS Synchronous Transport Signal. Synchronous Transport Signal level 1. Basic building block signal of SONET, operating at 51.84 Mbps. Faster SONET rates are defined as STS-*n*, where *n* is a multiple of 51.84 Mbps. *See also SONET*.

stub area In OSPF, an area through which, or into which, AS external advertisements are not flooded.

subnet mask Number of bits of the network address used for host portion of a Class A, Class B, or Class C IP address.

summary link advertisement OSPF link-statement advertisement flooded throughout the advertisement's associated areas by area border routers to describe the routes that they know about in other areas.

sysid System identifier. Portion of the ISO nonclient peer. The sysid can be any 6 bytes that are unique throughout a domain.

System and Switch Board *See SSB*.

T

TACACS + Terminal Access Controller Access Control System Plus. Authentication method for validating users who attempt to access the router using Telnet.

TCP Transmission Control Protocol. Works in conjunction with Internet Protocol (IP) to send data over the Internet. Divides a message into packets and tracks the packets from point of origin to destination.

ToS Type of service. The method of handling traffic using information extracted from the fields in the ToS byte to differentiate packet flows.

traffic engineering Process of selecting the paths chosen by data traffic in order to balance the traffic load on the various links, routers, and switches in the network. (Definition from http://www.ietf.org/internet-drafts/draft-ietf-mpls-framework-04.txt.) *See also MPLS*.

transit area In OSPF, an area used to pass traffic from one adjacent area to the backbone or to another area if the backbone is more than two hops away from an area.

transit router In MPLS, any intermediate router in the LSP between the ingress router and the egress router.

transport mode An IPSec mode of operation in which the data payload is encrypted, but the original IP header is left untouched. The IP addresses of the source or destination can be modified if the packet is intercepted. Because of its construction, transport mode can be used only when the communication endpoint and cryptographic endpoint are the same. VPN gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. *See also tunnel mode*.

Triple-DES A 168-bit encryption algorithm that encrypts data blocks with three different keys in succession, thus achieving a higher level of encryption. Triple-DES is one of the strongest encryption algorithms available for use in VPNs.

tunnel Private, secure path through an otherwise public network.

tunnel mode An IPSec mode of operation in which the entire IP packet, including the header, is encrypted and authenticated and a new VPN header is added, protecting the entire original packet. This mode can be used by both VPN clients and VPN gateways, and protects communications that come from or go to non-IPSec systems. *See also transport mode*.

Tunnel PIC A physical interface card that allows the router to perform the encapsulation and decapsulation of IP datagrams. The Tunnel PIC supports IP-IP, GRE, and PIM register encapsulation and decapsulation. When the Tunnel PIC is installed, the router can be a PIM rendezvous point (RP) or a PIM first-hop router for a source that is directly connected to the router.

type of service *See ToS*.

U

unicast Operation of sending network traffic from one network node to another individual network node.

UPS Uninterruptible power supply. Device that sits between a power supply and a router (or other piece of equipment) the prevents undesired power-source events, such as outages and surges, from affecting or damaging the device.

V

vapor corrosion inhibitor *See VCI*.

VCI Vapor corrosion inhibitor. Small cylinder packed with the router that prevents corrosion of the chassis and components during shipment.

VCI Virtual circuit identifier. 16-bit field in the header of an ATM cell that indicates the particular virtual circuit the cell takes through a virtual path. Also called a *logical interface. See also VPI*.

virtual circuit identifier *See VCI*.

virtual link In OSPF, a link created between two routers that are part of the backbone but are not physically contiguous.

virtual path identifier *See VPI*.

virtual private network *See VPN*.

Virtual Router Redundancy Protocol *See VRRP*.

VPI virtual path identifier. 8-bit field in the header of an ATM cell that indicates the virtual path the cell takes. *See also VCI*.

VPN virtual private network. A private data network that makes use of a public TCP/IP network, typically the Internet, while maintaining privacy with a tunneling protocol, encryption, and security procedures.

VRRP Virtual Router Redundancy Protocol. On Fast Ethernet and Gigabit Ethernet interfaces, allows you to configure virtual default routers.

W

wavelength-division multiplexing — *See WDM*.

WDM — Wavelength-division multiplexing. Technique for transmitting a mix of voice, data, and video over various wavelengths (colors) of light.

WFQ — Weighted Fair Queuing.

weighted round-robin — *See WRR*.

WRR — Weighted round-robin. Scheme used to decide the queue from which the next packet should be transmitted.

Part 7
Index

Index

Index

Symbols

A

B

C

D

Index

Index of Statements and Commands